Salient object detection and segmentation in videos

Qiong Wang

THESE DE DOCTORAT DE

L'INSA RENNES
COMUE UNIVERSITE BRETAGNE LOIRE

ECOLE DOCTORALE N° 601

Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Signal, Image, Vision*

Par

Qiong WANG

Salient object detection and segmentation in videos

Thèse présentée et soutenue à Rennes, le 09 MAI 2019
Unité de recherche : Univ Rennes, INSA Rennes, CNRS, IETR (Institut d'Electronique et de Télécommunication de Rennes), UMR6164
Thèse N° : 19ISAR 07 / D19 - 07

Rapporteurs avant soutenance :

Frederic DUFAUX
Directeur de recherche CNRS – CentraleSupelec
Guangtao ZHAI
Professeur - université de Shanghai Jiao Tong

Composition du Jury :

Président :
Didier COQUIN
Professeur des universités - université de Savoie Mont Blanc

Membres du jury :
Frederic DUFAUX
Directeur de recherche CNRS – CentraleSupelec
Guangtao ZHAI
Professeur - université de Shanghai Jiao Tong
Olivier LE MEUR
Maître de conférences HDR - université de Rennes 1
Directeur de thèse :
Kidiyo KPALMA
Professeur des universités - INSA Rennes
Co-encadrante de thèse :
Lu ZHANG
Maître de Conférences - INSA Rennes

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

INSA RENNES

Intitulé de la thèse :

Salient object detection and segmentation in videos

Qiong WANG

En partenariat avec :

ACKNOWLEDGEMENT

I would like to express my sincere gratitude towards my thesis advisors, Prof. Kidiyo KPALMA and Prof. Lu ZHANG. Thanks for providing me the opportunity to work with them, discussing the problems during the research, reviewing my reports, papers and thesis carefully, giving me many suggestions, encouragement and support.

I would like to thank my committee members, Prof. Didier COQUIN, Prof. Frederic DUFAUX, Prof. Guangtao ZHAI and Prof. Olivier LE MEUR. Thanks for their helpful comments and suggestions.

I also thank Prof. Sylvain GUEGAN for the helpful comments during intermediate defense.

My appreciation also goes to my colleagues from the laboratory VAADER team of IETR, INSA Rennes for the pleasure time. I would also like to acknowledge the people for sharing their knowledge.

I would like to express my special grateful to China Scholarship Council (CSC) for the financial supporting. I would like to thank my husband, parents and brother for their love.

RÉSUMÉ ÉTENDU

Introduction

Le système visuel humain a une capacité efficace à reconnaître facilement des régions d'intérêt dans des scènes complexes, même si les régions ciblées ont des couleurs ou des formes similaires à l'arrière-plan. La détection d'objets saillants (SOD) vise à détecter l'objet saillant qui attire le plus l'attention visuelle. La réponse d'un système SOD est une carte de saillance dans laquelle chaque pixel est étiqueté par une valeur réelle prise dans l'intervalle [0,1] pour indiquer sa probabilité d'appartenir à un objet saillant. Plus la valeur est élevée, plus la saillance est élevée.

En fonction de l'objectif visé, les approches existantes peuvent être classées globalement en deux catégories: les approches basées image et les approches basées vidéo. Les approches basées image modélisent le processus de vision en fonction de l'apparance de la scène. Le système visuel humain étant sensible aux mouvements, les approches basées vidéo détectent l'objet saillant en utilisant des indices, à la fois, du domaine spatial que du domaine temporel et deviennent de plus en plus populaires. Dans ce travail, nous nous concentrons sur les approches basées vidéo. Ce sujet a montré beaucoup d'intérêt notamment pour des applications exploitant l'attention humaine, telle que la conduite autonome [98], l'évaluation de la qualité, surveillance militaire, etc.

En conduite autonome, l'un des principaux problèmes est de garantir la robustesse de reconnaissance des panneaux de signalisation. Ces panneaux sont généralement de couleurs vives et attirent facilement l'attention humaine. Les approches de détection d'objets saillants dans une vidéo permettent de détecter le panneau de signalisation dans une scène dynamique, ce qui contribue à améliorer la sécurité lors de la conduite autonome.

Dans l'évaluation de la qualité d'image, la sensibilité du système visuel humain à divers les signaux visuels est importante. La détection d'objets saillants et l'évaluation de la qualité d'image sont toutes deux liées à la façon dont le système de visuel humain perçoit une image; les chercheurs intègrent des informations de saillance à des

modèles d'évaluation de la qualité d'image visant à améliorer leur performance. Une méthode habituelle consiste à utiliser la saillance comme une fonction de pondération pour refléter l'importance (ou la saillance) de la région dans une image.

Une autre application peut être trouvée dans la surveillance militaire. Les objets tels que les humains, les voitures et les avions attirent généralement un grand intérêt et doivent être soigneusement observé. Pour capter l'évolution de ces objets spécifiques, le calcul de la saillance fournit un indice important pour localiser les objets cibles.

Les méthodes de calcul de saillance basées vidéo insistent uniquement sur l'étiquetage de chaque pixel de l'image vidéo en indiquant "saillant" ou "non saillant". Pour les scènes réelles, la région saillante détectée peut contenir plusieurs objets (voir Fig.R1 (b)). Décomposer une région saillante en un ensemble d'objets différents est plus significatif et meilleur pour la compréhension de la vidéo. La Fig.R1 (c) montre la segmentation sémantique d'objets vidéo saillants [42] où tous les objets de même étiquette sémantique sont regroupés sous cette étiquette. Sur la Fig.R1 (d), on peut voir la segmentation semi-supervisée utilisant un étiquettage manuelle initial pour faire la segmention à travers la vidéo. L'assistance humaine est adoptée pour définir les objets d'intérêt qui sont généralement délimités dans la première image de la séquence. En propageant les étiquettes définies manuellement sur le reste de la séquence de la vidéo, l'instance de l'objet d'intérêt est segmentée dans l'ensemble de la séquence vidéo. La segmentation semi-supervisée d'objets vidéo peut être considérée comme un problème de suivi, mais avec le masque en sortie. Dans la carte en sortie, les pixels sont regroupés en plusieurs ensembles auxquels sont attribués une identité cohérente à l'objet: les pixels d'un même ensemble appartiennent au même objet.

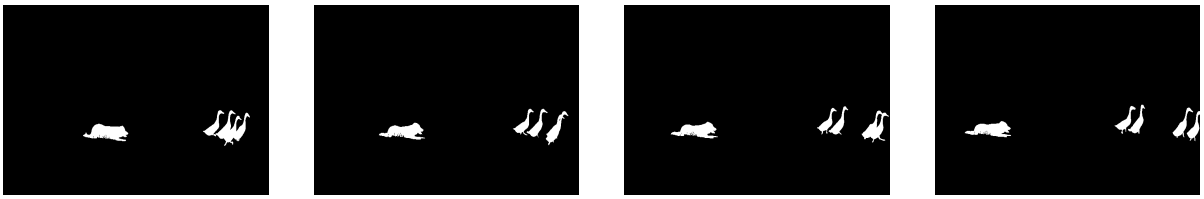
Ce dernier type de segmentation s'avère plus attractif mais n'a pas encore été complètement étudié et donc laisse de la place pour la recherche. Ainsi cette thèse s'est également intéressée à la segmentation semi-supervisée d'objets vidéo.

Notions de base sur la détection d'objets vidéo saillants

Lors de la création de jeux de données, de longues vidéos sont collectées par des volontaires ou sélectionnées à partir de sites Web de partage de vidéos comme Youtube. Ensuite, la fixation visuelle humaine est collectée pour une séquence vidéo d'entrée. A l'aide d'un système de eye-tracking, les participants aux expériences visualisent tous les courts vidéo-clips et les points de leur fixations sont enregistrés. Puis, tous les

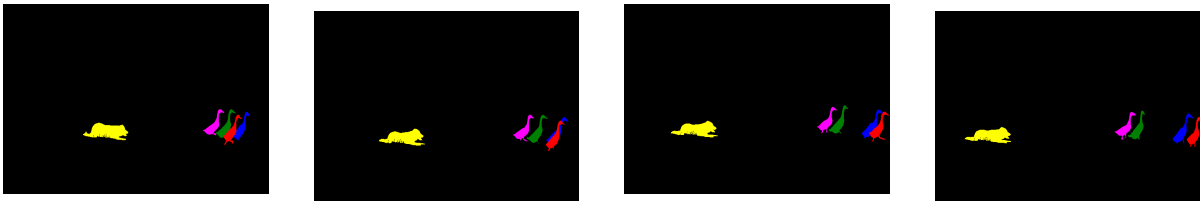
Première image Images d'entrée

(a)



(b) Vérité terrain (GT) d'objets vidéo saillants

(c) GT d'objets saillants basée sémantique



Etiquetage manuel GT pour la segmentation semi-supervisée de l'objet vidéo

(d)

Figure R1. Comparaison entre la détection d'objets saillants, la ségmentation sémantique d'objets saillants et la segmentation semi-supervisée d'objets saillants

masques d'objets sont annotés manuellement dans chaque image par les participants. Enfin, l'objet vidéo saillant est défini à l'échelle de la vidés entière: l'objet qui conserve les densités de fixation les plus élevées tout au long de la vidéo est sélectionné comme

objet saillant et la vérité-terrain est ainsi générée.

Dans le cadre de cette étude, cinq jeux de données ont été exploitées : VOS [49], Freiburg-Berkeley Motion Segmentation (FBMS) [8, 63], Fukuchi [27], DAVIS 2016-val [68] et DAVIS-2017-val [69]. Diverses métriques sont utilisées pour mesurer la similarité entre la carte de saillance générée (SM) et la vérité-terrain (GT). Les mesures couramment utilisées [6] sont: erreur absolue moyenne (MAE), courbe de précision-rappel (P-R), mesure de F-measure, rappel et précision.

Techniques traditionnelles de déctection d'objets vidéo saillants

Selon les techniques utilisées, les méthodes de détection d'objets vidéo saillants peuvent être grossièrement scendées en deux catégories: méthodes les traditionnelles et les méthodes utilisant l'apprentissage profond.

Dans cette étude, une nouvelle méthode traditionnelle de détection des objets saillants dans les vidéos est proposée. Les méthodes traditionnelles de détection d'objet basées sur l'a priori de l'arrière-plan peuvent rater des régions saillantes lorsque l'objet saillant touche les bords de l'image. Pour résoudre ce problème, nous proposons pour détecter la totalité de l'objet saillant d'ajouter les bordures virtuelles. Un filtre guidé est ensuite appliqué sur la sortisaillance temporelle en intégrant les informations de bordure spatiale pour une meilleure détection des objets saillants du bord. Enfin, une carte de saillance spatio-temporelle globale est obtenue en combinant la carte de saillance spatiale et la carte de saillance temporelle en fonction de l'entropie. Les principales contributions sont:

- une technique basées sur la notion de bordure virtuelle est proposée pour détecter un objet saillant connecté au bord de l'image,
- un filtre sensible aux contours est introduit pour fusionner les contours spatiaux avec les informations temporelles afin d'améliorer les contours des objets saillants,
- une nouvelle façon de décider du niveau de confiance de la carte de saillance spatiale et de la carte de saillance temporelle par le calcul de l'entropie et l'écart-type.

La Fig.R2 montre un des exemples de cartes de saillance générées à l'aide de la nouvelle approche VBGF exploitant les bords virtuels et le filtre guidé pour la détection d'objets vidéos saillants et la vérité-terrain (GT) correspondante.

Images d'entrée

Vérité terrain

Cartes de saillance générées

Figure R2. Exemples de cartes de saillance générées avec la méthode proposée (VBGF).

Revue des méthodes utilisant l'apprentissage profond pour la détection des objets saillants dans les vidéos

Ces dernières années, les méthodes d'apprentissage profond (ou deep-learning) ont considérablement amélioré la détection des objets saillants dans les vidéos. C'est un sujet important et il reste encore beaucoup à explorer. Il est donc intéressant de se faire une idée globale, sur les méthodes existantes, qui pourrait ouvrir la voie à des

travaux futurs. Les méthodes basées sur l'apprentissage profond peuvent atteindre des performances élevées, mais elles sont largement dépendantes des jeux de données d'apprentissage. Il est donc nécessaire de tester la généricité des méthodes de l'état de l'art en effectuant des comparaisons expérimentales sur différents jeux de données publics. Ainsi, nous donnons un aperçu des développements récents dans ce domaine et comparons les méthodes correspondantes à ce jour. Les principales contributions sont:

- un apperçu des méthodes récentes d'apprentissage profond pour la détection d'objets saillants dans les vidéos,
- un classement des méthodes de l'état de l'art ainsi que leur architecture,
- une étude expérimentales comparative pour tester la généricité des méthodes de l'état de l'art à travers des expérimentations sur des bases de données publiques.

Afin de montrer comment la performance d'un modèle traditionnel de détection d'objets vidéo saillants peut être encore améliorée en intégrant une méthode d'apprentissage profond, une méthode étendue (VBGFd) est proposée. C'est la version élargie de la méthode traditionnelle VBGF proposée intégrant la technique de deep-learning. La Fig.R3 montre exemples de cartes de saillance générées par la méthode proposée (VBGFd).

Méthode deep-learning pour la segmentation semi-supervisée de l'objet vidéo

Dans le domaine de segmentation semi-supervisée de l'objet vidéo, la technique de déformation de masque, qui adapte (recale) le masque de l'objet cible en fonction du flux de vecteurs entre images consécutives, est largement utilisée pour extraire l'objet cible. Le gros problème de cette approche est que la carte déformée générée n'est pas toujours d'une grande précision, l'arrière-plan ou d'autres objets pouvant être détectés à tort comme étant l'objet cible. Pour remédier à ce problème, nous proposons une méthode SWVOS, qui utilise la sémantique de l'objet cible comme guide lors du processus de recalage. Le calcul du taux de confiance de déformation détermine d'abord la qualité de la carte déformée générée. Ensuite, une sélection de la sémantique est

Images entrées

Vérité terrain

Cartes de saillance générées

Figure R3. Exemples de cartes de saillance générées par la méthode proposée (VBGFd).

introduite pour optimiser la carte à faible taux de confiance, où l'objet cible est identifié, à nouveau, à l'aide de l'étiquette sémantique de l'objet cible. Les contributions sont:

- une méthode est proposée pour déterminer le niveau de confiance des cartes de recalés,

- la sémantique des objets est introduite pour filtrer les objets du premier plan appartenant à des classes différentes de celle de l'objet prédéfini.

La Fig.R4 montre certaines cartes de segmentation générées par l'approche proposée SWVOS.


Première image et étiquettes générées manuellement

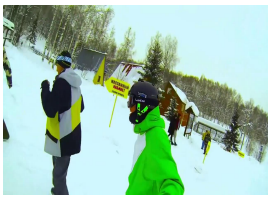
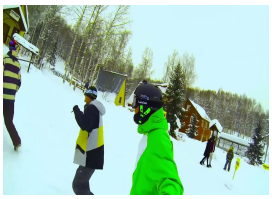
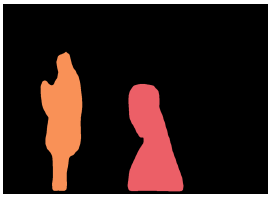
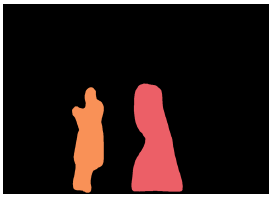
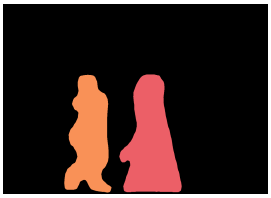
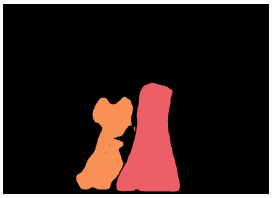



Image d'entrée

Résultats de la segmentation

Figure R4. Exemples de segmentations obtenues à l'aide de la méthode proposée (SWVOS).

Conclusion et perspectives

Cette thèse porte sur les problèmes de détection d'objets vidéo saillants destinée à la séparation des objets saillants de l'arrière-plan dans chaque image d'une séquence vidéo et les problèmes de segmentation semi-supervisée de l'objet vidéo qui visent à attribuer une dentité d'objet cohérente à chaque pixel de chaque image d'une séquence vidéo. Nous avons proposé une méthode traditionnelle de détection d'objets vidéo saillants et une revue des méthodes deep-learning pour la détection d'objets vidéo saillants. Nous avons égalment introduit une extension de la méthode traditionnelle proposée pour y intégrer le deep-learning et une méthode de deep-learning pour la segmentation semi-supervisée de l'objet vidéo. Les approches proposées ont été évaluées sur les jeux de données publics à grande échelle et difficiles. Les résultats expérimentaux obtenus montrent que les approches proposées donnent des résultats

satisfaisants.

Certains travaux futurs peuvent être dérivés des analyses précédentes: utiliser des représentations plus riches les achitectures de deep-learning qui pourraient améliorer les performances l'approche VBGFd proposé; entrainer les réseaux de deep-learning pour la fusion de cartes de saillance pour améliorer l'approche VBGF proposée qui peut faillir quand les saillance temprelles et spatiales ne sont pas suffisamment nettes. On peut également, envisager à employer plus d'indices de saillance vidéo prenant en compte l'attention visuelle humaine. Il sera aussi intéressant d'explorer davantage les aspects temporels et spatio-temporels qui permettraient d'assurer une detection de saillance tout le long de la vidéo. essayez des réseaux faiblement supervisés. Enfin, on peut envisager d'explorer les réseaux faiblement supervisés. En effet, les modèles supervisés améliorent les performances de détection, mais reposent sur un jeu de données volumineux d'apprentissage. Les modèles faiblement supervisés qui ne demandent de grandes masses de données retient l'attention et constituent un sujet d'intérêt pour l'avenir.

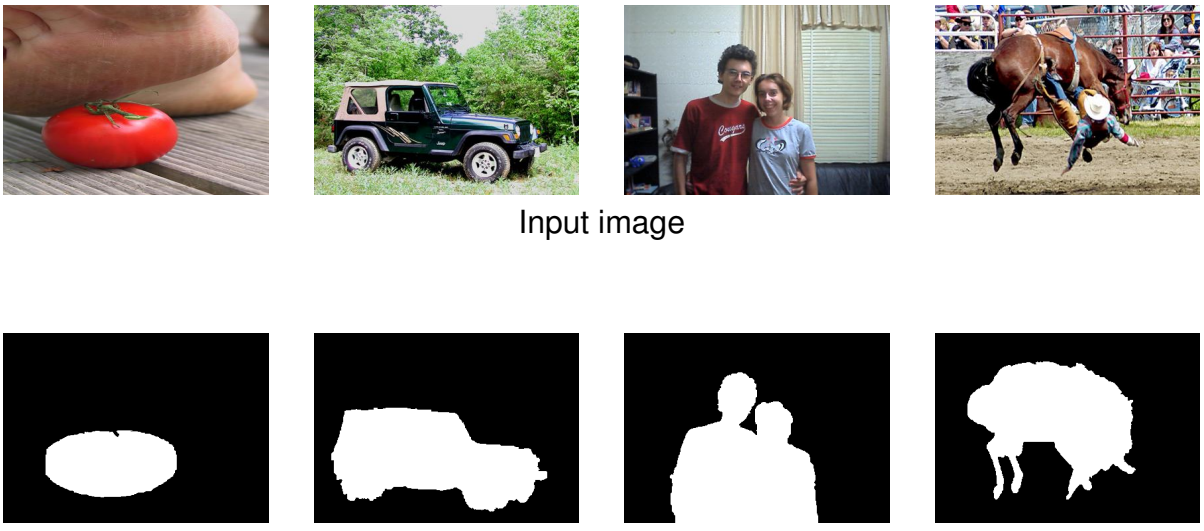
TABLE OF CONTENTS

INTRODUCTION

Background

The human vision system has an effective ability to easily recognize regions of interest from complex scenes, even if the focused regions have similar colors or shapes as the background. Salient object detection (SOD) aims to detect the salient object that attracts the most the visual attention. The output of the SOD is a saliency map, in which each pixel is labeled by a real value within the range of [0,1] to indicate its probability of belonging to a salient object. Higher value represents higher saliency.

According to the goal of detection, existing approaches can be broadly classified into image SOD or video SOD, which are illustrated in Fig.0.1 and Fig.0.2 respectively.

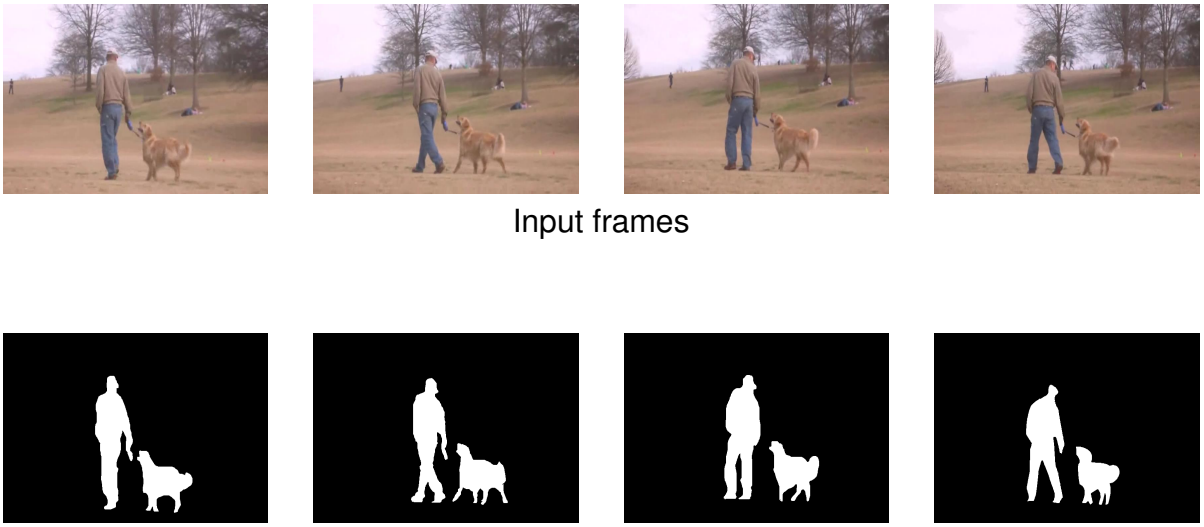


Input image

Ground truth (GT)

Figure 0.1: Examples of image SOD.

Image SOD models the visual processing based on the appearance of the scene. Since the human vision system is sensitive to motions, video SOD detects the salient object using cues in both spatial domain and temporal domain. In this present work,



Input frames

GT of the corresponding frame

Figure 0.2: Examples of video SOD.

we focus on video SOD. This topic has gained much attention for its wide applications, especially where the task is driven by the human attention, such as autonomous driving [98], quality assessment, military surveillance, etc.

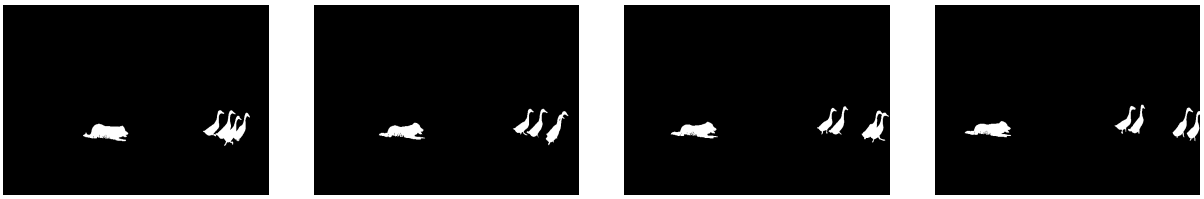
In autonomous driving, one of the biggest issue is to ensure the robustness of road signs recognition. Road signs are generally in brightly colors and easily catch the human attention. The video salient object detection is good at discovering the road sign in a dynamic scene, which helps to improve the safe during autonomous driving.

In image quality assessment, the sensitivity of the human visual system to various visual signals is important. As salient object detection and image quality assessment are both related to how human vision system perceives an image, researchers incorporate saliency information to image quality assessment models aiming at improving their performance. One usual way is to adopt salient object detection as a weighting function to reflect the importance region in an image.

Another application can be found in military surveillance. The objects such as humans, cars and airplanes usually attracts a lot of interests and need to be carefully observed. To grasp the trend of these specific objects, video salient object detection provide a useful cue to localize target objects.

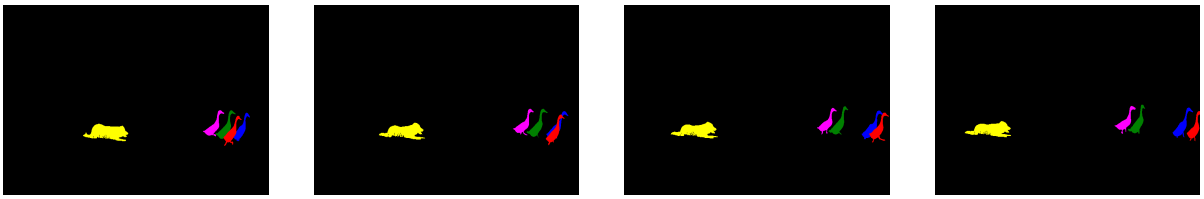
For video SOD (see Fig.0.3 (b)), in which the pixels with high value represent salient objects, and the pixels with zero value represent background. There is trend to solve this problem from traditional method to deep-learning based method. Tradi-

(a) Input frames



(b) GT of video SOD

(c) GT of video semantic salient object segmentation



(d) GT of video object instance segmentation

Figure 0.3: A comparison of video SOD, video semantic salient object segmentation and video object instance segmentation.

tional methods usually detect the salient object based on hand-crafted features and prior assumptions, while deep-learning methods detect the salient object based on deep representations which are learned from training datasets with provided ground truth. For a given database, deep-learning methods have a better performance than many recent traditional methods. But it should be trained with huge and rich training

datasets, which is impossible for some applications where the available data is small. Traditional methods do not suffer from such limitation. Therefore, we firstly focus on video SOD based on traditional method, i.e., detecting salient object based on prior assumption. Deep-learning methods attract large attention for its high accuracy and efficiency. We secondly focus on video SOD based on deep learning methods.

The aforementioned video SOD methods put emphasis on only labeling each pixel in the video frame to be “salient” or “non-salient”. For real-world scenes, the detected salient region may contain multiple objects (see Fig.0.3 (b)). Decomposing the detected region into different objects is more meaningful and is better for video understanding. Video semantic salient object segmentation [42], as show in Fig.0.3 (c), segments the salient region based on the semantic label, in which the salient objects belonging to the same semantic label are grouped together. From Fig.0.3 (d), in the output map of video object instance segmentation, the pixels are grouped into multiple sets and assigned to consistent object IDs. Pixels within the same set belong to the same object.

Video object instance segmentation attracts more interests and has not been fully investigated. We address the problem of assigning consistent object IDs to objects instance. One popular way for video object instance segmentation is called as Semi-supervised video object segmentation. Human-guidance is adopted to define the objects that people want to segment. It is usually delineated in the frame that the object appears in the first time. By propagating the manual labels to the rest of the video sequence, the object instance is segmented in the whole video sequences. Semi-supervised video object segmentation can be regarded as a tracking problem but with the mask output.

Overview of the thesis

The thesis is organized as in Fig.0.4. Chapter 1 introduces the preliminary knowledge about saliency detection. Chapter 2 is dedicated to a proposed traditional approach for video SOD, and an overview of recent deep-learning based methods and an extended model are proposed for video SOD in Chapter 3. In Chapter 4, a semi-supervised video object segmentation approach is proposed. Chapter 5 concludes the thesis and gives some perspectives for future work. The following parts give a briefly introduction of each chapter, in order to lead readers to better understanding the content.

Chapter 1 introduce the basic knowledge for video SOD and semi-supervised video

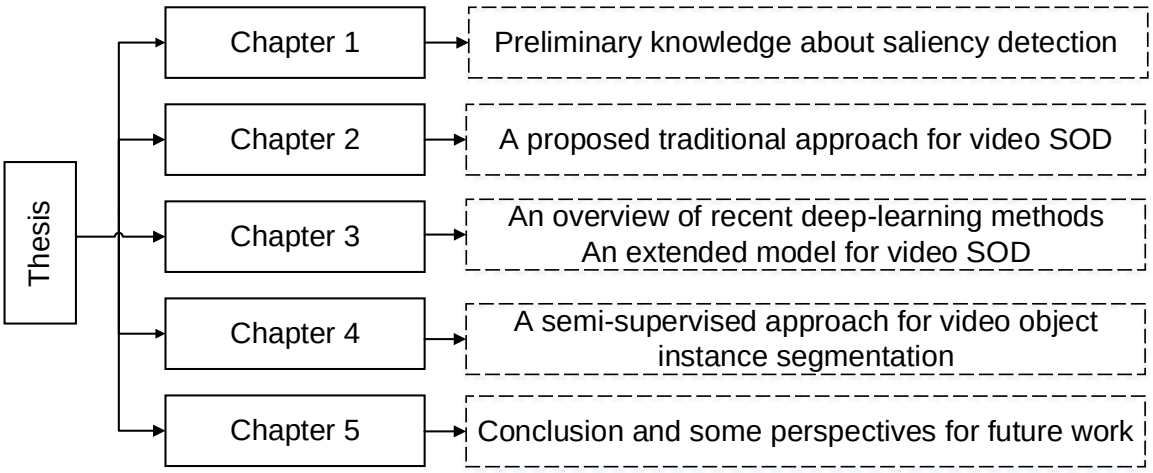


Figure 0.4: Overview of the thesis.

object segmentation:

- A description of the dataset building.
- A list of popularly used datasets.
- A introduction of widely used evaluation metrics.

Chapter 2 presents a novel traditional method (Virtual Border and Guided Filter-based salient object detection for videos (VBGF)) for solving challenging problems in existing traditional methods:

- A virtual border-based technique for detecting the salient object connected to frame borders using the distance transform.
- An edge-aware filter to fuse the spatial edge with the temporal information for enhancing salient object edges.
- A new way to decide the confidence level of the spatial saliency map and the temporal saliency map by computing Entropy and Standard deviations.

Fig.0.5 shows some saliency maps generated by VBGF and the corresponding GT.

Chapter 3 puts emphasis on the analysis of the state-of-the-art methods in video SOD based on deep-learning techniques, which mainly concludes:

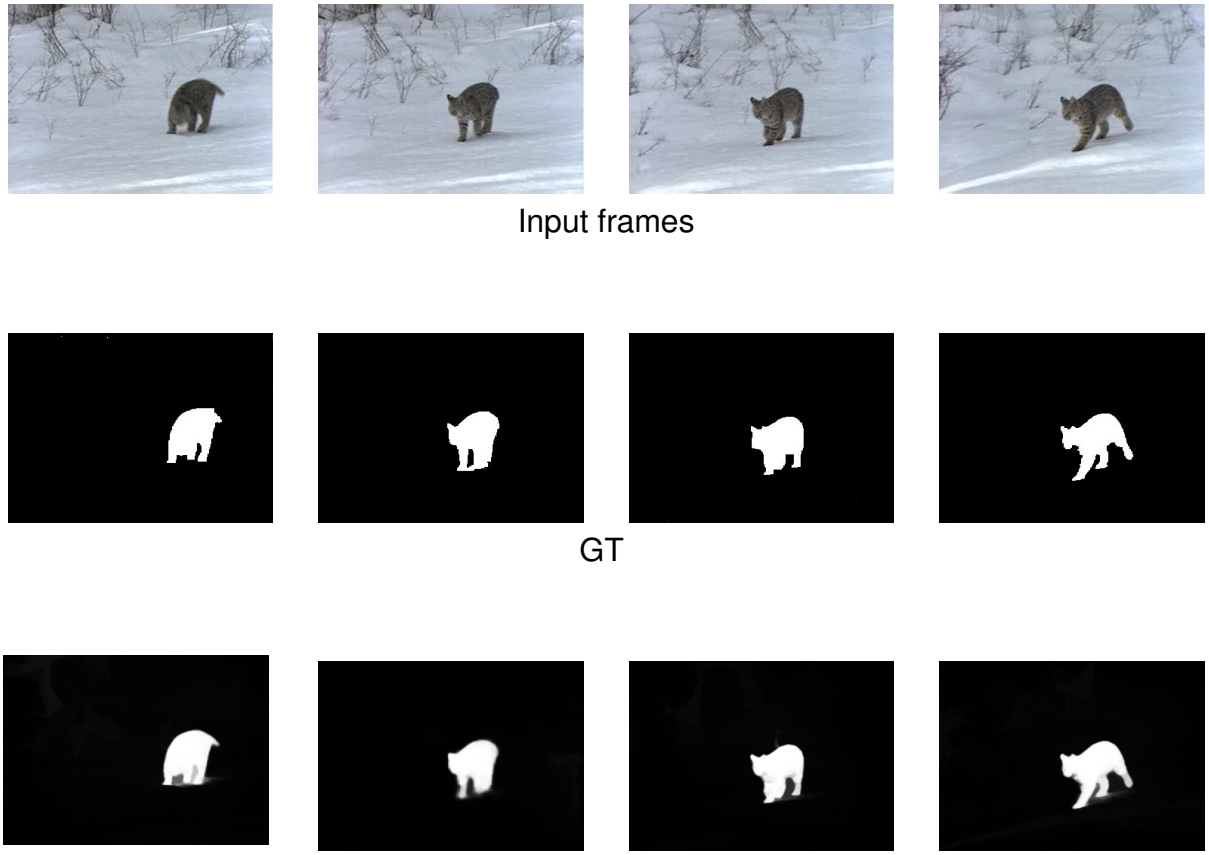


Figure 0.5: Some examples of saliency maps generated by the proposed VBGF.

- An overview of recent deep-learning based methods for salient object detection in videos is presented;
- A classification of the state-of-the-art methods and their frameworks is provided.
- Experiments are made to test the generality of state-of-the-art methods through experimental comparison on different public datasets.
- An extension of the VBGF (VBGFd) by integrating a deep-learning technique is proposed and the performance is evaluated.

Fig.0.6 shows some examples of saliency maps generated by the VBGFd.

Chapter 5 proposes a Semantic-guided warping for semi-supervised video object segmentation (SWVOS) to address the semi-supervised video object segmentation problem:

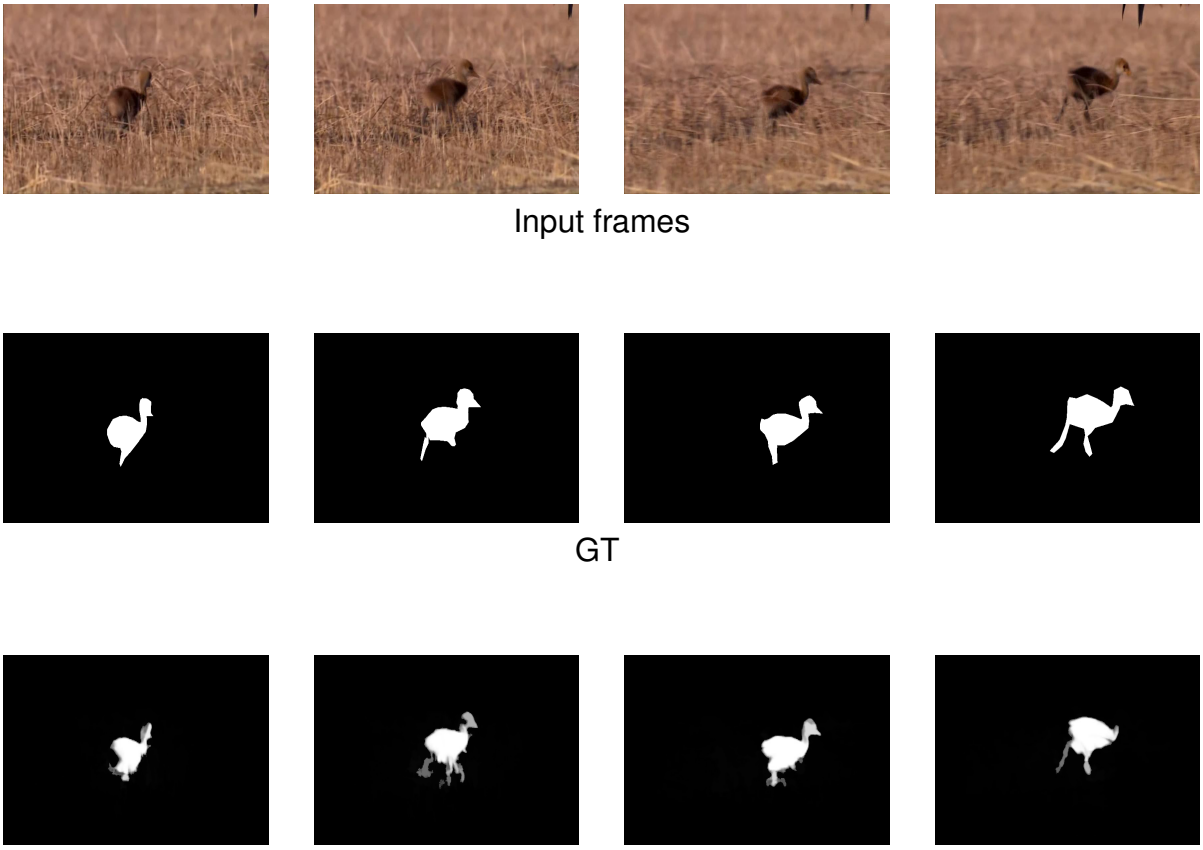


Figure 0.6: Some examples of saliency maps generated by VBGFd.

- A selection method is proposed to decide the confidence level of the warped maps.
- Object semantic is introduced to filter foreground object belonging to the class which is different from the class of the pre-defined object.

Fig.0.7 shows some segmentation maps generated by the proposed approach SWVOS.

The first frame and its manual labels

Input frames

Segmentation results

Figure 0.7: Some examples of segmentation maps generated by SWVOS.

CHAPTER 1

BASIC KNOWLEDGE

Chapter 1 firstly introduces the procedure of dataset building for video SOD in Section 1.1. Then, benchmarking datasets built in recent years are introduced in Section 1.2. Thirdly, evaluation metrics are finally listed in details in Section 1.3.

1.1 Procedure of dataset building

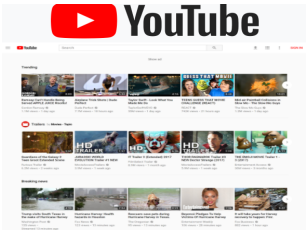
This section introduces the constructing of the video SOD dataset [49]. In the procedure of dataset building, long videos are collected by volunteers or selected from video-sharing websites like Youtube. Then short clips are randomly sampled to keep the clips containing objects in most frames. Then the human fixation is collected for an input video sequence. Subjects participate in the eye-tracking experiments are required to free-view all video short clips and their fixations are recorded. Thirdly, all object masks are manually annotated by subjects for each frame. Finally, the video salient object is defined at the scale of whole videos: the object that keeps the highest fixation densities throughout a video is selected to be the salient object, and the GT is generated. The procedure as shown in Fig.1.1.

1.2 Benchmarking datasets

This section reviews the most popular datasets for video SOD and semi-supervised video object segmentation, respectively.

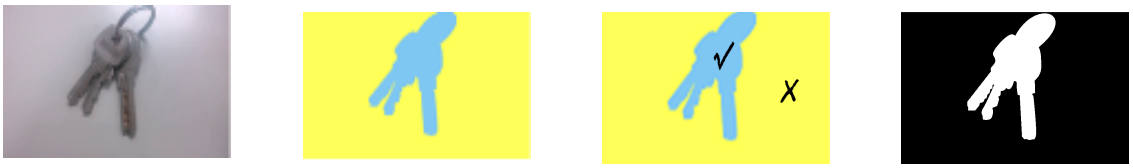
The VOS [49] dataset is a recently published large dataset for SOD in videos, which is based on human eye fixation. These videos are grouped into two subsets: 1) VOS-E contains easy videos which usually contain obvious foreground objects with many different types of slow camera motion. 2) VOS-N contains normal videos which contain complex or highly dynamic foreground objects, and dynamic or cluttered background.

Video collection by volunteers



Video collection from video-sharing websites

Human fixation on different frames in a video sequence



Input frame | Annotated masks | Selection | GT

Figure 1.1: Examples of dataset building.

Due to the limited number of large-scale datasets designed for SOD in videos, existing methods usually use datasets from highly related domains like the datasets hereafter.

The FBMS dataset [8, 63] is designed for moving object segmentation. Moving objects attract large attention and thus can be regarded as salient objects in videos. As in the methods [17], we use the 30 test videos for test and only evaluate the result of frames which are provided the ground truth. It includes different cases (such as “the salient object touches the frame border” in sequence “marple7”, “the salient object is very similar to the background” in “dog01” and “cars1”, “multiple objects” in “cars5_20”, “horses04_0400” and “people2_10” or “the background is complex” in “cats01”).

The Fukuchi [27] dataset, designed for video object segmentation, includes 10 sequences. Since most objects have distinct colors or are very dynamic, they can be con-

sidered as salient objects. The salient object touches the frame border in most video sequences, such as in “DO01_013” all the salient objects touch the frame border and in “M07_058”, “DO01_055” and “DO02_001” part of salient objects touch frame border. All tested methods hardly detect the salient object for one video sequence “BR128T”. As in [14], this sequence “BR128T” is excluded in the test.

The DAVIS 2016-train-val dataset [68] is a popular video dataset for video foreground segmentation. It is divided into two splits: the training part used for training only and the validation part for the inference. It is widely used for SOD in videos, because of the foreground properties (most of the objects in the video sequences have distinct colors, which can be regarded as salient objects). The DAVIS-2017-train-val dataset [69] is a recently published video dataset. It is divided into two splits: training and validation. It is mainly an extension of DAVIS-2016 dataset.

The detailed information of these datasets are listed in Table 1.1.

Table 1.1: Comparison between various test datasets.

Dataset	Numbers			Resolution
	Sequence	Frame	GT	
VOS	200	116103	116103	[408,800]
VOS-E	97	49206	49206	[408,800]
VOS-N	103	66897	66897	[448,800]
FBMS-test	30	13860	720	[350,960]
Fukuchi	10	740	740	[352,288]
DAVIS 2016-val	20	1376	1376	[480,854]
DAVIS 2017-val	30	1999	1999	[480,854]

The YouTube-VOS dataset [96] is a recently published and the largest dataset with high resolution for semi-supervised video object segmentation. It is the most challenging dataset, and it contains three sets: Train, Val and Test. It has the total number 197,272 of object annotations. For the Test set, it contains 508 video sequences with the first-frame ground truth provided. 65 categories of objects in the Test set appear in Train set, which are called as “seen objects”; and 29 categories of objects in the Test set do not appear in Train set, which are called as “unseen objects”.

To illustrate datasets, some examples are given in Fig.1.2

Figure 1.2: Some examples are given for each dataset.

1.3 Evaluation metrics

For video SOD, various metrics are used to measure the similarity between the generated saliency map (SM) and GT. The more commonly used metrics are:

- Mean Absolute Error (MAE): computed as the average absolute difference between all pixels in SM and GT. A smaller MAE value means a higher similarity and a better performance.

$$\mathrm{MAE} = \frac{1}{h1 \times w1} \sum_{i=1}^{h1 \times w1} |\mathrm{GT}(i) - \mathrm{SM}(i)| \tag{1.1}$$

  where h_1 is the frame height, w_1 is the frame width.

- Precision-Recall (P-R) curve [6]: SM is normalized to $[0, 255]$ and converted to a *binary mask (BM)* via a threshold that varies from 0 to 255. For each threshold, a pair of (Precision, Recall) values are computed which are used for plotting P-R curve. The curve closest to the upper right corner (1.0, 1.0) corresponds to the best performance.

$$\mathrm{Precision} = \frac{|\mathrm{BM} \bigcap \mathrm{GT}|}{|\mathrm{BM}|}, \quad \mathrm{Recall} = \frac{|\mathrm{BM} \bigcap \mathrm{GT}|}{|\mathrm{GT}|} \tag{1.2}$$

- F-measure: used to evaluate the global performance:

$$\mathrm{F-measure} = \frac{(1 + \beta^2) \times (\mathrm{Precision} \times \mathrm{Recall})}{(\beta^2 \times \mathrm{Precision} + \mathrm{Recall})} \tag{1.3}$$

  β^2 is often set to 0.3. A higher F-measure mean a better performance.

Note that the benchmark [49] adopts an adaptive threshold (computed as the minimum value between “maximum pixel value of saliency map” and “twice the average values of saliency map”) to convert the saliency map to a binary mask, and the calculates of metrics (MAE, Precision, Recall and F-measure). A higher F-measure, Precision and Recall values mean a better performance.

For video SOD evaluation, the metrics values are firstly computed over each video, and secondly computed the mean values over all videos in each dataset.

For semi-supervised video object segmentation, Region Similarity *J* and Contour Accuracy *F* [68] are used to measure the similarity between the generated segmentation map (M) and the ground truth (GT). Region Similarity *J* is defined as the intersection-over-union of *M* and GT. Contour Accuracy *F* is computed by the contour-based precision *Pc* and recall *Rc*.

$$J = \frac{|M \bigcap \mathrm{GT}|}{|M \bigcup \mathrm{GT}|} \quad F = \frac{2P_c R_c}{P_c + R_c} \tag{1.4}$$

A larger *J* value and a larger *F* value mean a better performance. For the overall evaluation, the final measure is the average of four scores: *J* for seen categories, *J* for unseen categories, *F* for seen categories and *F* for unseen categories.

CHAPTER 2

TRADITIONAL TECHNIQUES FOR SALIENT OBJECT DETECTION IN VIDEOS

In this chapter, Section 2.1 gives an overview of state-of-the-art methods dedicated to video salient object detection. Section 2.2 describes some issues existing in recent works. Section 2.3 presents the proposed method in detail. In Section 2.4, we conduct comparison experiments to evaluate the performance of the proposed method. Section 2.5 concludes the chapter.

2.1 An overview of state-of-the-art methods

A large number of approaches have been developed for detecting video salient objects based on traditional methods. Various low-level saliency cues are exploited for detection and different fusion ways are used to fuse the spatial and the temporal information together.

2.1.1 Classification based on low-level saliency cues

For video SOD, we propose to classify low-level saliency cues into three categories: prior assumption, foreground object and moving object.

Saliency cues: prior assumption

Contrast prior, spatial distribution prior, background prior, boundary connectivity prior, center prior and objectness prior [28] are most popular. Specifically, color contrast prior is mostly used in early works to capture the uniqueness in a scene. Chen *et al.* [15] obtain the motion saliency via contrast computation. Chen *et al.* [14] compute the color contrast and the motion contrast respectively. Spatial distribution prior implies that the

wider a color is distributed in the image, the lesser likely a salient object contains this color; background prior assumes that a narrow border of the image is the background region; boundary connectivity cue is based on the assumption that most of the image boundaries will not contain parts of the salient object: the boundary connectivity score of a region according to the ratio between its length along the image border and the spanning area of this region; center prior assumes that a salient object is more likely to be found near the image center, so it is usually used as a weighting coefficient on saliency maps; objectness prior leverages object proposals as the salient object cue; focusness prior assumes that a salient object is often photographed in focus to attract more attention.

For the saliency value computation, distance transform, graph-based, structured matrix decomposition, etc. are recently used measures. The features are usually extracted in pixel-level or superpixel-level. For superpixel-level, the image is decomposed by using superpixel segmentation which groups similar pixels and generates compact regions. For distance transform, the saliency value is computed as the shortest distance from each pixel or superpixel to seed pixels. Seed pixels selection is the key of distance transform. Based on background prior, Wang *et al.* [91] consider the spatio-temporal edge map border as seed pixels. Yang *et al.* [100] consider the four borders as seed set individually. Xi *et al.* [93] select the spatio-temporal seeds based on boundary connectivity cue. For graph-based method: an image is over-segmented into superpixels and mapped to one single graph. The saliency value of each superpixel is then computed based on the similarity between connected nodes and the saliency related queries. For structured matrix decomposition [3], a matrix is decomposed into a low-rank matrix representing background and a sparse matrix identifying salient objects.

Saliency cues: foreground object

Video foreground object [23] which is separated from the background is another popular saliency cue for SOD in videos. Using foregroundness cue , Tu *et al.* [82] compute foreground weights to estimate saliency maps. Chen *et al.* [17] define the foreground potential and background potential based on reliable object region and background region. Chen *et al.* [14] assign high saliency value around foreground object. Aytekin *et al.* [1] extract the salient segments by applying a spectral foreground detection method. Kim *et al.* [39] detect the foreground salient objects. Guo *et al.* [28] separate the foreground object from the background to produce an initial saliency estimation.

Saliency cues: moving object

Moving objects [4, 58] usually attract largely the human attention. Temporal saliency is detected from motion information. The optical flow method is one of the most popular tools to extract the motion information effectively. The salient object can be detected using the optical flow vectors by removing redundant motion (i.e. global motion, including the camera movement or the background motion). For the redundant motion computation, Tu *et al.* [81] propose that if the percentage of motion magnitude greater than the half of the maximum motion magnitude is larger than 50%, the global motion exists. Luo *et al.* [56] set the major direction along x-axis (either positive x-axis or negative x-axis) and y-axis (either positive y-axis or negative y-axis) to be the global motion in optical flow vectors. Cassagne *et al.* [12] calculate the mean value of the magnitude and the orientation of optical flow vectors as the global motion. Decombas *et al.* [20] compute the average value of optical flow vector along x-axis and y-axis as the global motion. These methods only use the motion information between adjacent frames [105] to detect the salient object in temporal domain. However, the general idea of video salient object is that it has a coherent motion over time. It means that motion consistence need to be considered. Liu *et al.* [54] propagate motion saliency measures over video sequences. Zhou *et al.* [106] provide a bidirectional temporal propagation.

Fig. 2.1 shows the classification of the video SOD methods based on low-level cues.

2.1.2 Classification based on fusion ways

For video SOD, both spatial and temporal information can help the saliency detection. We propose to classify the existing methods into “Map fusion”, “Feature fusion” and “Hybrid fusion” methods. “Map fusion” firstly obtains the spatial saliency map and the temporal saliency map, and then combines them together. “Feature fusion” is to fuse the extracted spatial feature and extracted temporal feature together to give a spatio-temporal feature, which is used to generate the spatio-temporal saliency map. In order to employ more video saliency information, these two techniques are used together in “Hybrid fusion” recently.

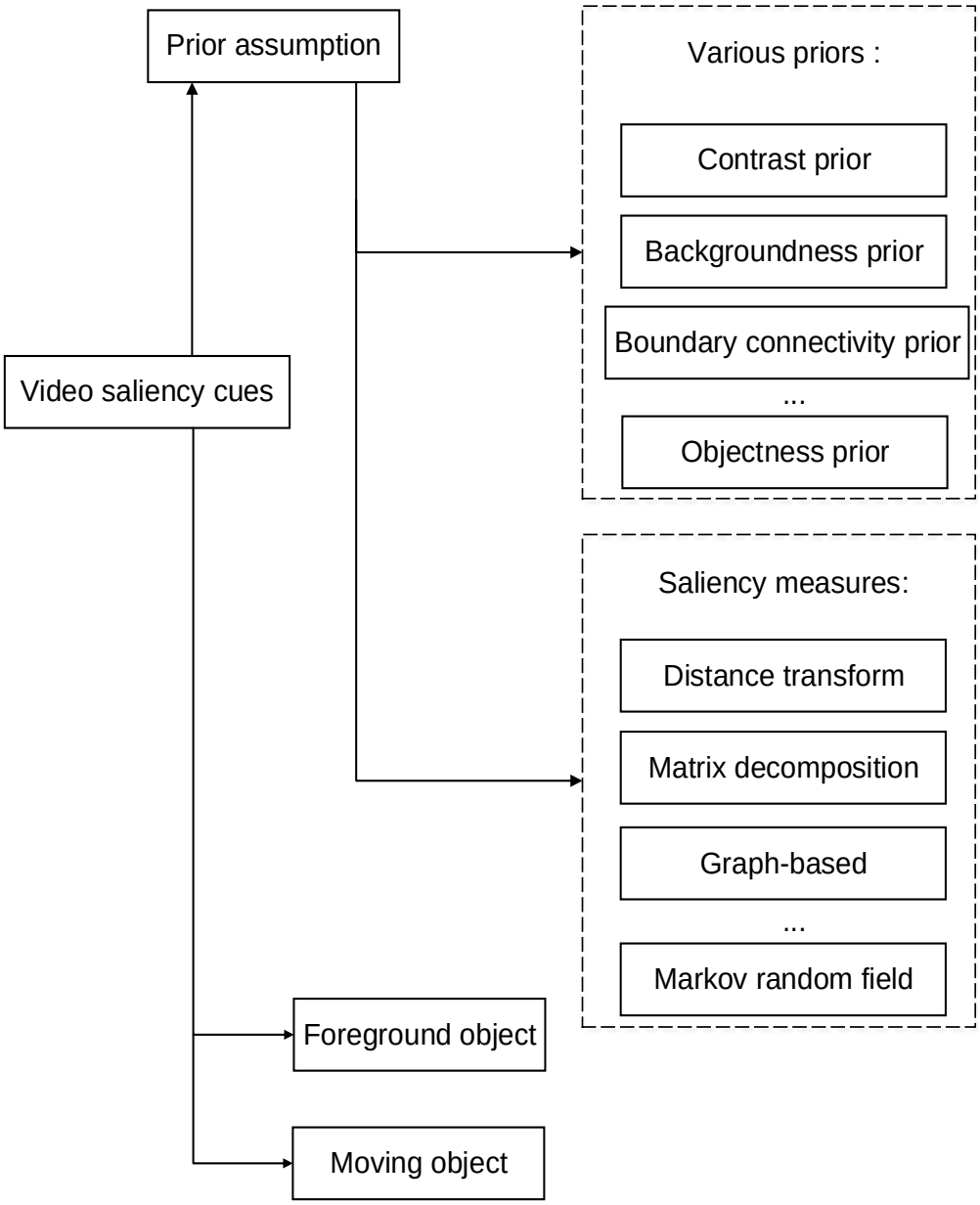


Figure 2.1: Methods classification based on low-level cues.

Map fusion

Fang *et al.* [26] give an adaptive weighted fusion rule with an uncertainty computation on both spatial and temporal saliency maps. Kannan *et al.* [37] propose a Max fusion. For each pixel, the fused saliency is the larger one between spatial saliency and temporal saliency. Duan *et al.* [22] combine these two saliency maps in a non-linear way, based on the assumption that spatially dissimilar and moving blocks are more visually attractive. Tu *et al.* [81] propose to equally weight both saliency maps in a linear way. Zhai *et al.* [102] propose a dynamic fusion technique where temporal gaze attention is dominate over the spatial domain when large motion contrast exists, and vice versa. Tu *et al.* [82] generate two types of saliency maps based on a foreground connectivity saliency measure, and exploit an adaptive fusion strategy. Yang *et al.* [100] propose a confidence-guided energy function to adaptively fuse spatial and temporal saliency maps. Ramadan *et al.* [71] apply the pattern mining algorithm to recognize spatio-temporal saliency patterns from two saliency maps.

Feature fusion

Wang *et al.* [89] and Wang *et al.* [88, 91] detect the salient object from the fused spatio-temporal gradient field. Guo *et al.* [28] select a set of salient proposals via a ranking strategy. Li *et al.* [49] fuse the spatial and temporal channel to generate saliency maps, and then use saliency-guided stacked autoencoders to get the final saliency map. Bhattacharya *et al.* [3] use a weighted sum of the sparse spatio-temporal features. Chen *et al.* [15] obtain the motion saliency map with spatial cue, then use k-Nearest Neighbors-histogram based filter and Markov random field to eliminate the dynamic backgrounds.

Hybrid fusion

Kim *et al.* [39] generate the spatio-temporal map based on the theory of random walk with restart, which use the temporal saliency map as the restarting distribution of the random walk. Liu *et al.* [54] obtain the spatio-temporal saliency map using temporal saliency propagation and spatial propagation. Xi *et al.* [93] first get spatio-temporal background priors, and the final saliency value is the sum of appearance and motion saliency. Zhou *et al.* [105] generate the initial saliency map, and propose localized

estimation to generate the temporal saliency map, and deploy the spatio-temporal refinement to get the final saliency map, which is then used to update the initial saliency map. Chen *et al.* [14] get the temporal saliency map to facilitate the color saliency computation. Chen *et al.* [17] detect the motion cues and spatial saliency map to get the motion energy term, which are combined with some constraints and formulated into the optimization framework. Chen *et al.* [13] employ contrast cue, devise a Markov random field solution and learn multiple nonlinear feature transformations to detect the video salient object detection. Fig. 2.2 shows the classification of the video SOD methods based on fusion ways.

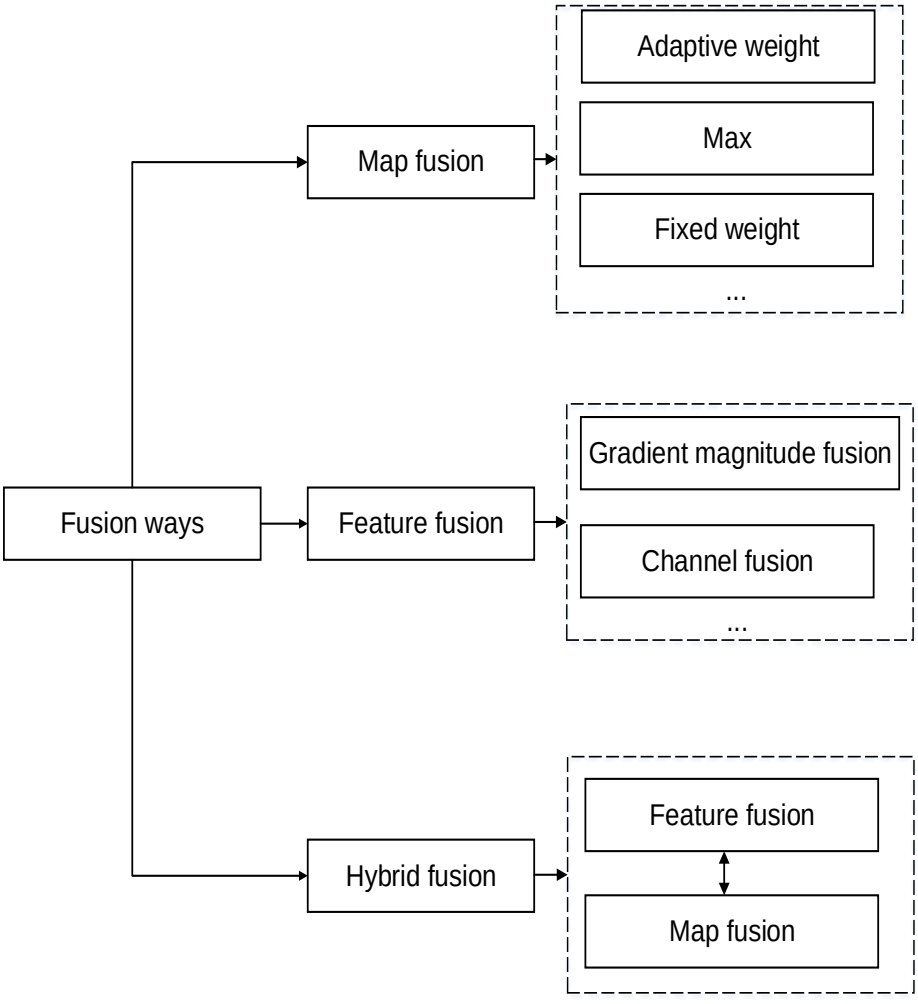


Figure 2.2: Methods classification based on fusion ways

2.2 Introduction of some existing issues

In this section, a brief overview of some existing issues related to our work is given. The "background prior" [6] is widely used in SOD approaches based on traditional techniques. A narrow border of the image is assumed to be the background region. When the salient object pixels appear in the border, their saliency values are set incorrectly to zeros. Besides, video SOD detects the salient object from both spatial domain and temporal domain. How to combine these two saliences together during the detection is complex.

2.2.1 Background prior

Based on this assumption, the distance transform [72] has been widely used for saliency computation. Traditionally, the distance transforms measure the connectivity of a pixel and the seed set using different path cost functions. Since background regions are assumed to be connected to image borders, the border pixels are initialized as the seed set and the distance transform detects a pixel's saliency by computing the shortest path from the pixel to the seed. The shorter the shortest path is, the higher the saliency is. In the background prior, all the border pixels are regarded as background. Thus, in the distance transform, all the border pixels are set to be seed and their saliency values are thus zeros. This is not true if the object of interest appears in the frame border.

Based on "background prior", Zhang *et al.* [103] propose a salient object detection method based on the Minimum barrier distance transform. Combined with the raster scanning, a fast iterative Minimum barrier distance transform algorithm (FastMBD) detects the initial image saliency. In addition, the region possessing a very different appearance from image boundary is highlighted. For each image boundary region, the mean color and the color covariance matrix are calculated using the pixels inside this boundary region. Then the intermediate image boundary contrast map is obtained based on the Mahalanobis distance from the mean color. The final boundary contrast map is got from the four intermediate image boundary contrast maps. After the initial saliency map is integrated with the image boundary contrast map, a morphological smoothing step, a centeredness map and a contrast enhancement operation are used as post-processing operations. Fig. 2.3 gives an example, in which Fig. 2.3 (b) shows the initial saliency result using FastMBD and Fig. 2.3 (c) presents the final result. Fig.

2.3 (c) is improved but the lower part of the person is not detected, since it touches the border of the frame.

(a) (b) (c)

Figure 2.3: FastMBD15 [103]. (a) Input image, (b) Minimum barrier distance transform with the Raster Scan, (c) Final result. (Figures are copied from the published paper [103])

Tu *et al.* [80] combine the Minimum barrier distance transform with a minimum spanning tree. Instead of finding the shortest distance, they search the shortest path in the minimum spanning tree. The minimum spanning tree is constructed by avoiding edges with large color difference between adjacent pixels. To ensure the detection of the salient object that touches the frame border, the boundary color dissimilarity measure is used. They first divide the boundary into three groups according to their color values and then the intermediate pixel-wise color dissimilarity map of each group is calculated using the Mahalanobis distance. The final color dissimilarity map is the weighed sum of three intermediate color dissimilarity maps. In the post-processing, a boundary dissimilarity map, a pixel location dependent masking and an adaptive contrast enhancement are used. Fig. 2.4 gives an example to show the intermediate results.

Jiang *et al.* [36] propose a saliency detection via absorbing Markov chain on an image graph model. It measures image saliency by using the similarity of the absorbed time of each transient node with the background absorbing nodes (the image border). It considers both the edge weights on the path and the spatial distance when computing the absorbed time, so the object that is different from or far from the background absorbing nodes can be highlighted. The homogeneous background region in the image center may not be effectively suppressed. The saliency map is updated using a weighted absorbed time. Fig. 2.5 compares the results without update processing and

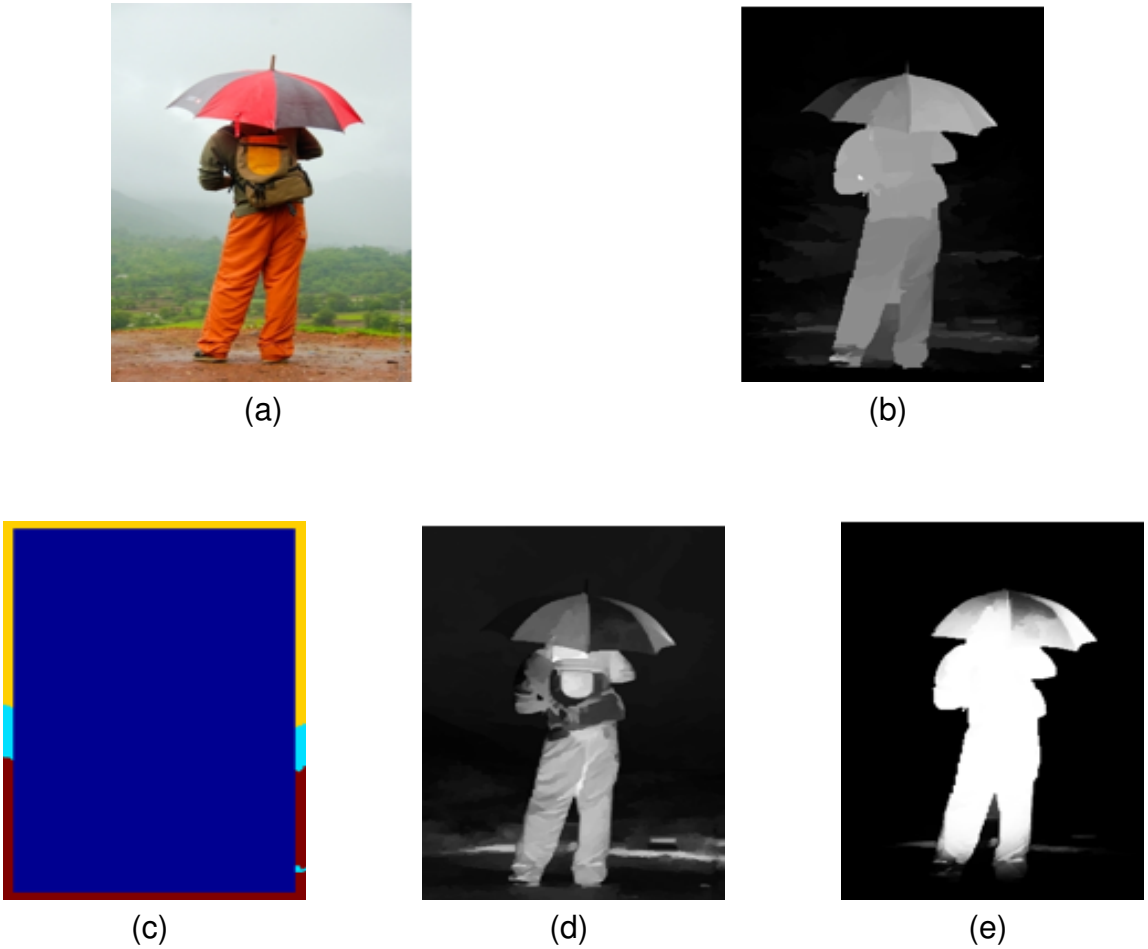
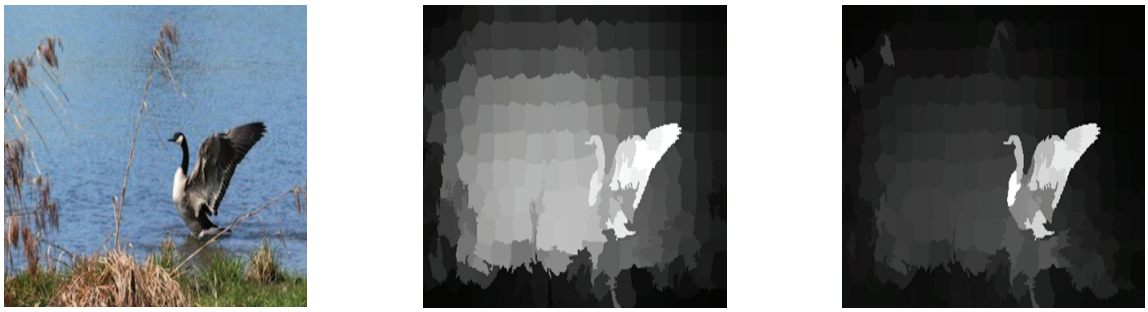


Figure 2.4: MST16 [80]. (a) Input image, (b) Minimum barrier distance transform with the minimum spanning tree, (c) Boundary index (the boundary is divided into three groups according to their color values), (d) Boundary dissimilarity map, (e) Final result. (Figures are copied from the published paper [80])

with update processing.



(a) (b) (c)

Figure 2.5: AMC13 [36]. (a) Input image, (b) Result without update processing, (c) Result with update processing. (Figures are copied from the published paper [36])

Frame GT

[80] [103] [36]

Figure 2.6: State-of-the-art saliency maps [36, 80, 103].

Though these methods [36, 80, 103] can alleviate this problem, but not enough. Fig.2.6 shows the saliency maps of these three methods on one example image.

2.2.2 Spatial and temporal information fusion

Various methods are proposed to decide the confidence weight for each saliency map.

The method in [89] fuses the superpixel color gradient magnitude and optical flow gradient magnitude into a spatio-temporal gradient field in a non-linear way. An exponential function is employed to emphasize the optical flow gradient magnitude. Then, the entire salient object is highlighted with the fused spatio-temporal edges. The local contrast and global contrast are introduced to highlight an entire object, and an energy function is used to encourage the spatio-temporal consistency. Fig. 2.7 illustrates the saliency estimation steps.

Wang *et al.* [88] fuses the static edge probability map, the superpixel segmentation result and the motion boundary into a spatio-temporal edge probability map. The spatio-temporal saliency map is obtained by computing the shortest geodesic distance from each superpixel to two adjacent frames borders. A skeleton abstraction step is further used to improve the performance. Fig. 2.8 gives an example to show the detailed intermediate results.

Liu *et al.* [54] first generate the temporal saliency map in the superpixel-level. Then, temporal saliency propagation is obtained using spatial appearance, and spatial propagation is performed via the temporal saliency map to obtain the spatio-temporal saliency maps. Fig. 2.9 presents the above steps in an example.

Kim *et al.* [39] use the temporal saliency map as the restarting distribution of the random walker. The spatial saliency is extracted via the transition probability matrix. The saliency maps of two domains are fused into a framework based on the theory of random walk with restart. Then the generated spatio-temporal map is used to update the temporal saliency distribution. Fig. 2.10 gives an example to compare the saliency maps generated by only using spatial information and employing temporal information as restarting distributions respectively.

Chen *et al.* [14] first employ contrast cue to get the low-level saliency. Then a Markov random field solution is devised to obtain the Pos region (salient), Neg region (non salient) and Unk region (undeterministic). Multiple nonlinear feature transformations are learned and help to assign saliency values to those Unk region. Finaly, spatio-temporal smoothness is enforced. Fig. 2.11 shows the steps of this method.

In complex scenes, existing methods still could not fully make use of detected saliency from the two domains. Some examples are shown in Fig.2.12. For video SOD models [39, 54, 89] which detect the salient object in spatial and temporal domains, the salient object are with blur edges. Thus, the fusion is still a challenging problem.

Facing these open issues, we propose a new video salient object detection algo-

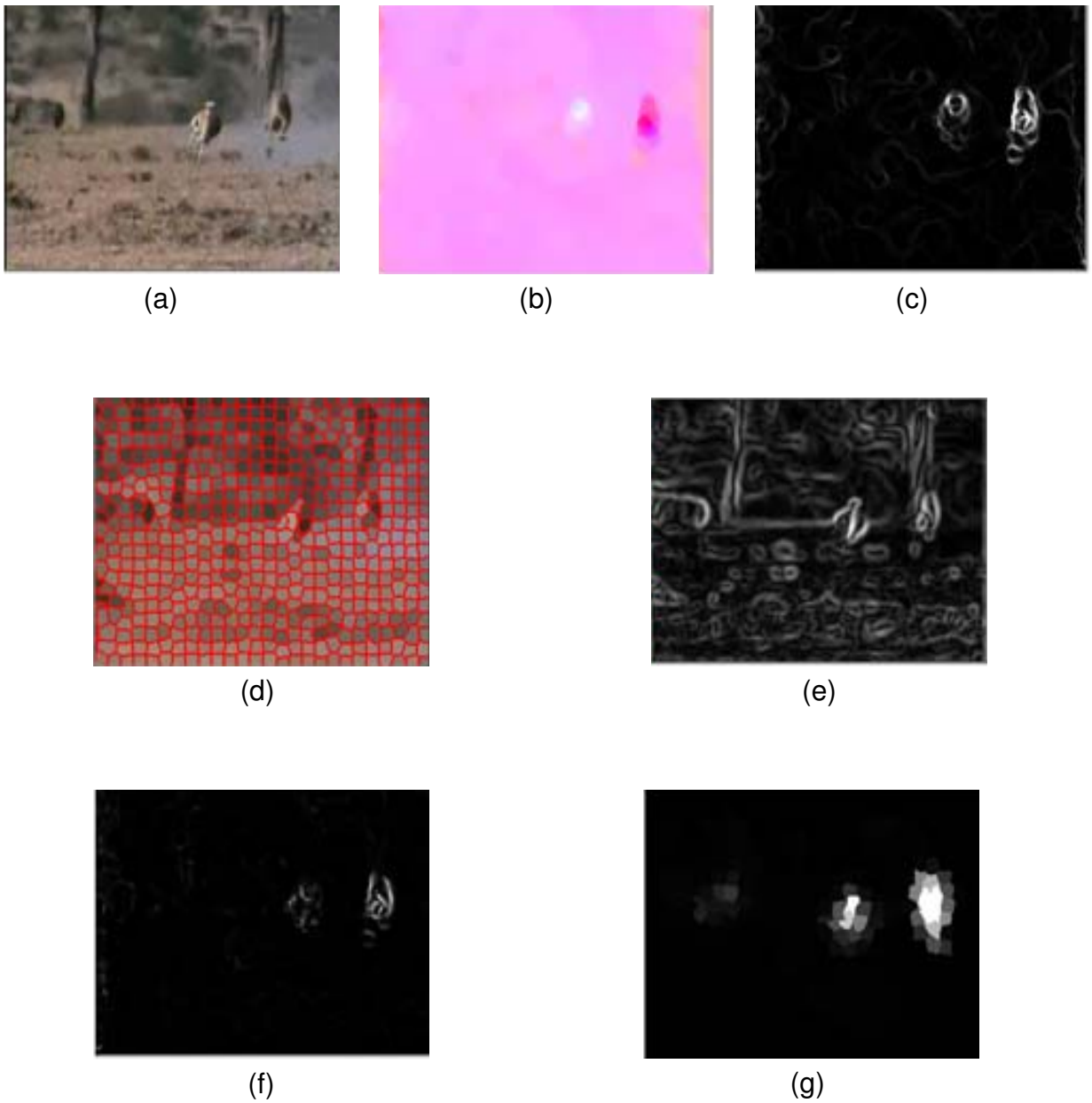


Figure 2.7: GF15 [89]. (a) Input frame, (b) Color optical flow map of the input frame, (c) Optical flow gradient magnitude, (d) Superpixel segmentation, (e) Gradient magnitude of (d), (f) Spatio-temporal gradient field by fusing (c) and (e) in a non-linear way, (g) Final result. (Figures are copied from the published paper [89])

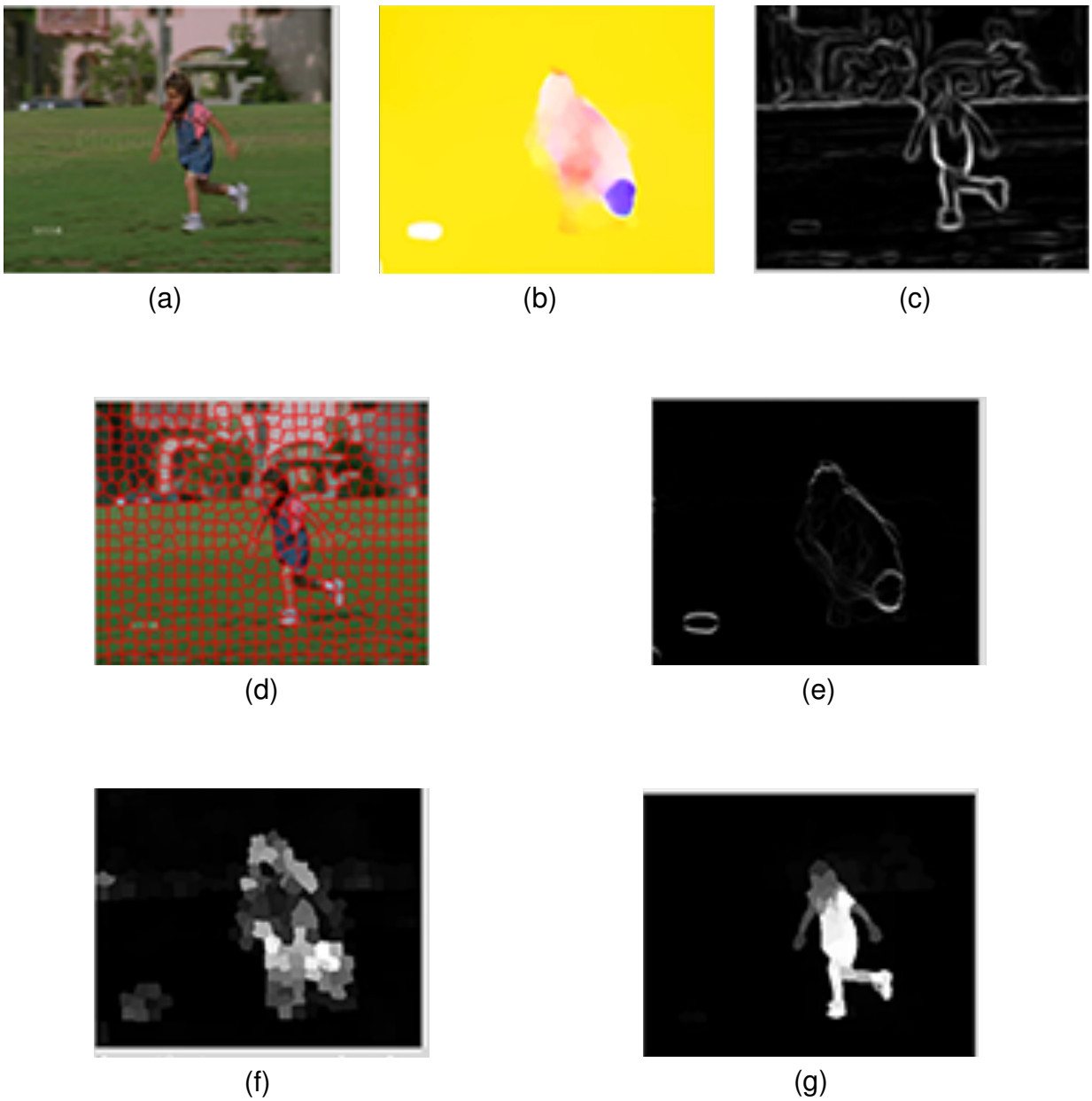


Figure 2.8: SAG15 [88]. (a) Input frame, (b) Color optical flow map of the input frame, (c) Static edge probability map, (d) Superpixel segmentation, (e) Motion boundary of (b) , (f) Spatio-temporal edge probability map by combining (c), (d) and (e), (g) Final result. (Figures are copied from the published paper [88])

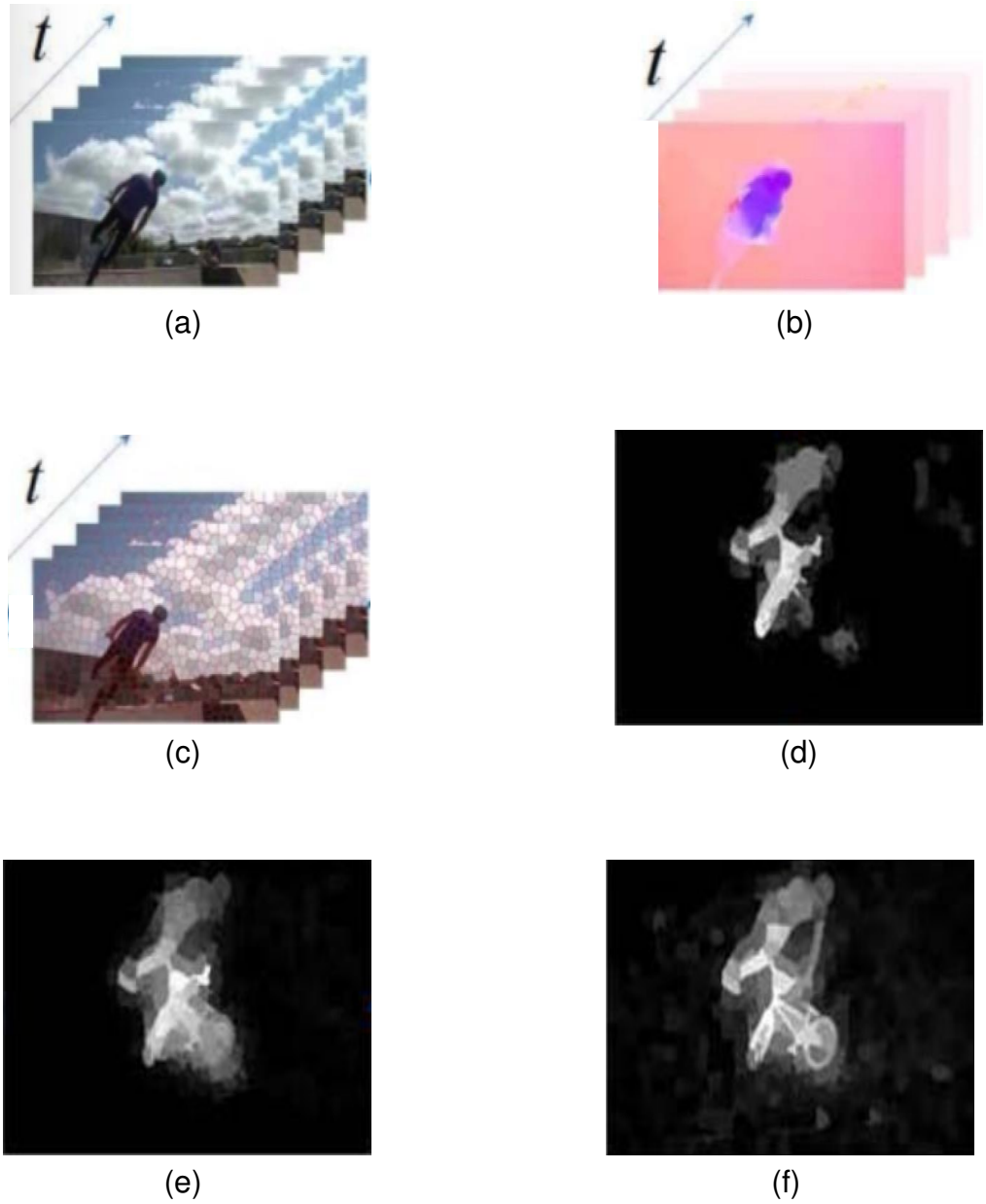
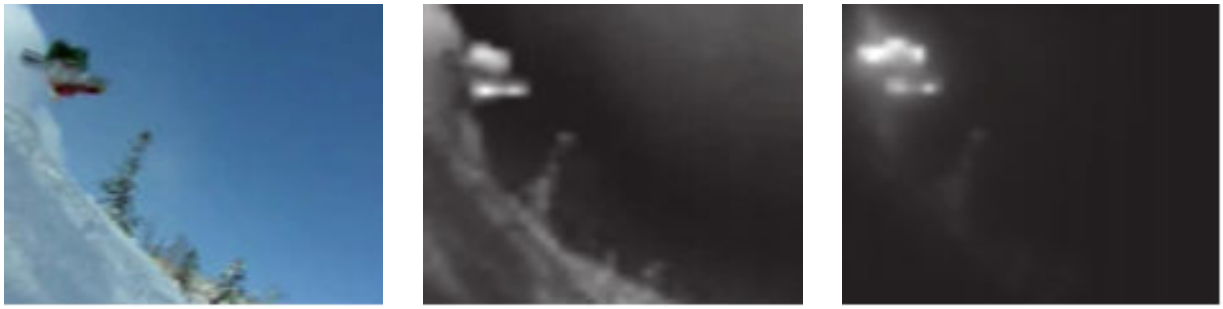


Figure 2.9: SGSP16 [54]. (a) Input frame, (b) Color optical flow map of the input frame, (c) Superpixel segmentation, (d) Graph based motion saliency, (e) Spatial propagation, (f) Final result. (Figures are copied from the published paper [54])



(a) (b) (c)

Figure 2.10: RWR15 [39]. (a) Input frame, (b) Saliency map generated by the random walk simulation without employing temporal information as restarting distributions, (c) Saliency map generated by the random walk simulation with employing temporal information as restarting distributions. (Figures are copied from the published paper [39])

rithm. We propose to detect the whole salient object via the adjunction of virtual borders from both spatial and temporal domains. A guided filter is then applied on the temporal information to integrate the spatial edge information for a better detection of the salient object edges. At last, a global spatio-temporal saliency map is obtained by combining the spatial saliency map and the temporal saliency map together according to the entropy.

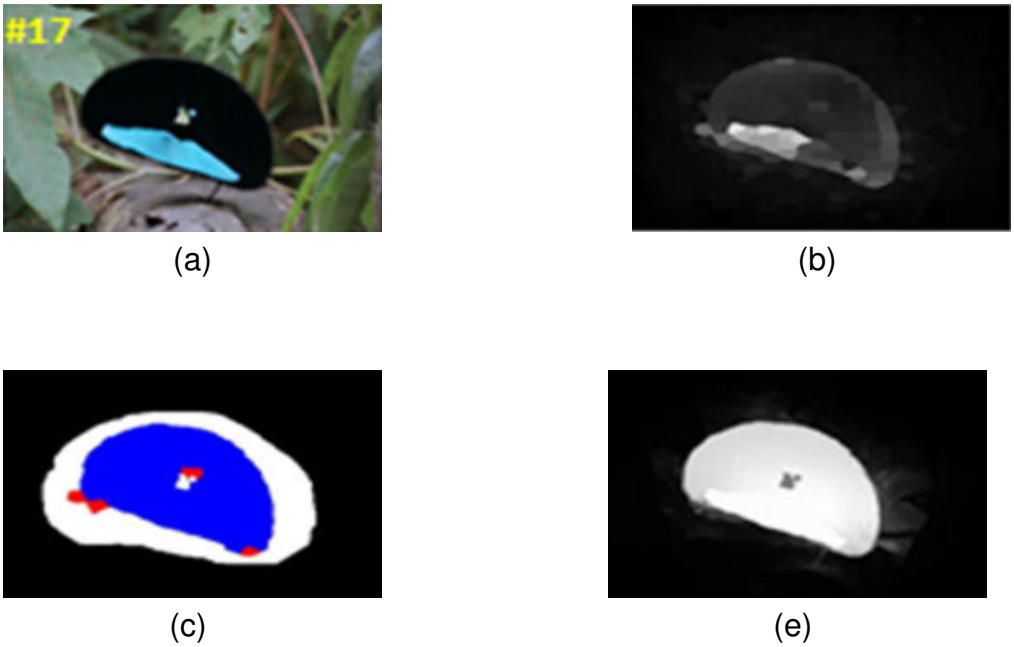


Figure 2.11: FD17 [14]. (a) Input frame, (b) Contrast-based saliency, (c) Pos region (salient) are denoted by blue color, Neg region (non salient) are denoted by red color and Unk region (undeterministic) are denoted by white color, (d) Final result. (Figures are copied from the published paper [14])

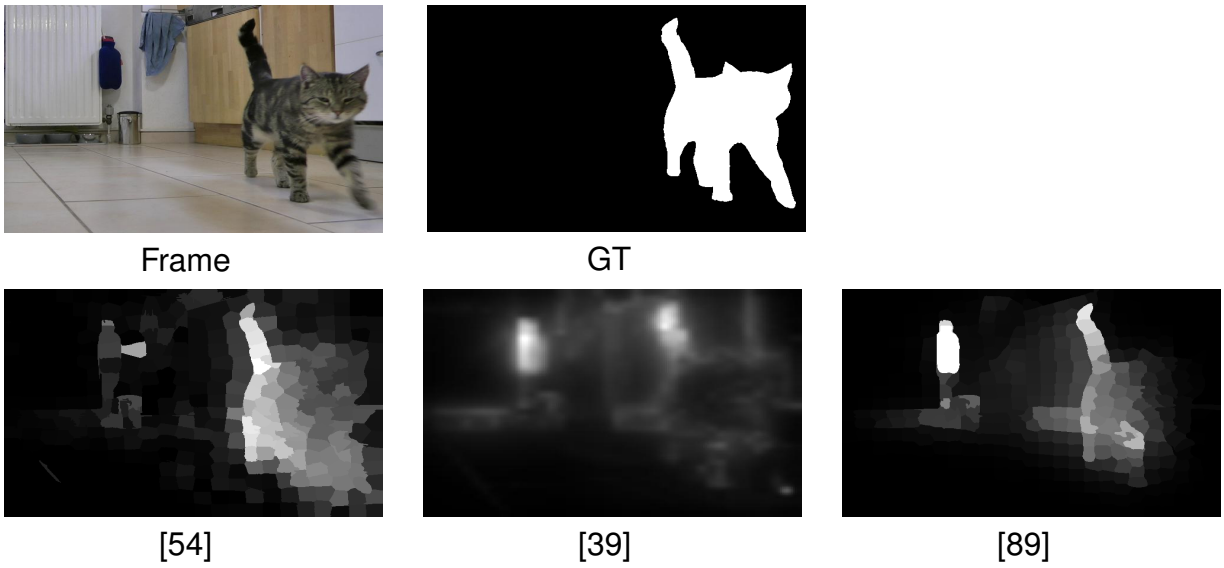


Figure 2.12: State-of-the-art saliency maps [39, 54, 89].

2.3 Virtual Border and Guided Filter-based (VBGF) algorithm

The block-diagram of the proposed VBGF method is shown in Fig.2.13.

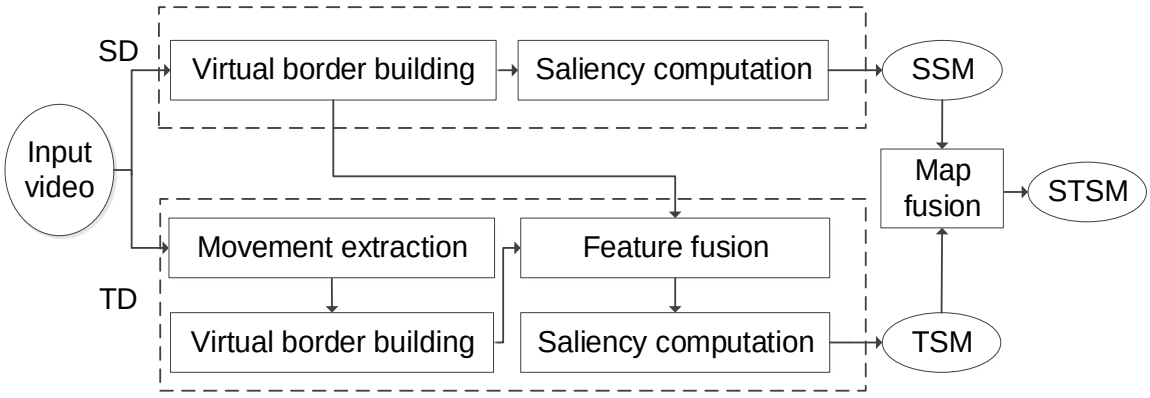


Figure 2.13: The proposed block-diagram. SD: Spatial saliency detection; SSM: Spatial saliency map; TD: Temporal saliency detection; TSM: Temporal saliency map; STSM: Spatio-temporal saliency map.

Given an input video sequence, in Spatial saliency detection (SD), the virtual border is built for each frame. Then, the saliency is computed to get the SSM. Secondly, in Temporal saliency detection (TD), the motion information is extracted from the input video. Then the virtual border building, the Feature fusion and the saliency computation are applied to obtain the TSM. At last, the two saliency maps are fused to get the STSM. The method is detailed in the following parts.

2.3.1 Spatial saliency detection

In this part, Spatial saliency detection (SD), the virtual border-based distance transform in spatial domain, is designed.

Virtual border building

Instead of using the frame border pixels as the seed set, we propose to add virtual borders around the original frame to obtain with-virtual-border frame. The virtual bor-

der, calculated using original frame border pixel values, is used to get the new seed set. Specifically, the virtual border is built in four steps (as shown in Fig.2.14): Frame Border Selection, Frame Border Division, Representative Pixel Selection and Virtual Border Padding.

a) Frame Border Selection: it may suppose that the salient object could be connected with two or more borders. However, from the existing video datasets we observe that: in usual cases if the salient object appears in the frame border, it is often connected with only one border. Here for the sake of simplicity of the presentation, we select one original frame border to build the virtual border by two steps.

In the first step, Fast iterative Minimum barrier distance transform algorithm (FastMBD) [103] is applied to frame α to obtain the map M as Eq (2.1).

$$M = \frac{1}{3}({M_1}' + {M_2}' + {M_3}'). \tag{2.1}$$

where ${M_1}'$, ${M_2}'$ and ${M_3}'$ are obtained respectively from three color channels of frame α in the CIELab color space. For each color channel I with the size of $h_1 \times w_1$, M' is generated as follows: if the pixel $x \in r_1$ (r_1 being the border of the frame α), its value in M' is set to 0. If pixel $x \in r_2$ (r_2 being the non-border of the frame), its value in M' is initialized as ∞. Two auxiliary maps τ and ψ are initialized by the pixel values in each channel of the original image. Let the 4-adjacent pixels around a pixel x in the region r_2 be x_{up} (up pixel), x_{left} (left pixel), x_{down} (down pixel) and x_{right} (right pixel). Using the update function, M' and the auxiliary maps are firstly updated in raster scan order, secondly updated in inverse raster scan order with $y \in \left\{x_{\mathrm{down}}, x_{\mathrm{right}}\right\}$, and thirdly updated in raster scan order again. The update function is shown as follows: if $M'(x) > O_y(x)$ ($y \in \left\{x_{\mathrm{up}}, x_{\mathrm{left}}\right\}$), $M'(x)$, τ_y and ψ_y are updated to $O_y(x)$, $\max\left\{\tau_y, I(x)\right\}$ and $\min\left\{\psi_y, I(x)\right\}$ respectively, where $O_y(x) = \max\left\{\tau_y, I(x)\right\} - \min\left\{\psi_y, I(x)\right\}$.

In the second step, the frame border nearest to the non-zero region in the map M is selected to build the virtual border. Here, the threshold δ is used to determine the non-zero region.

b) Frame Border Division: after one original border selected, the corresponding divided border is obtained from the original frame border (with width u). The DUB, the DDB, the DLB and the DRB are shown in the middle left part in Fig.2.14. The reason lying behind this division is that: the region in the frame corner is often connected with two borders and its feature is also related to these two borders. Thus, the irregular

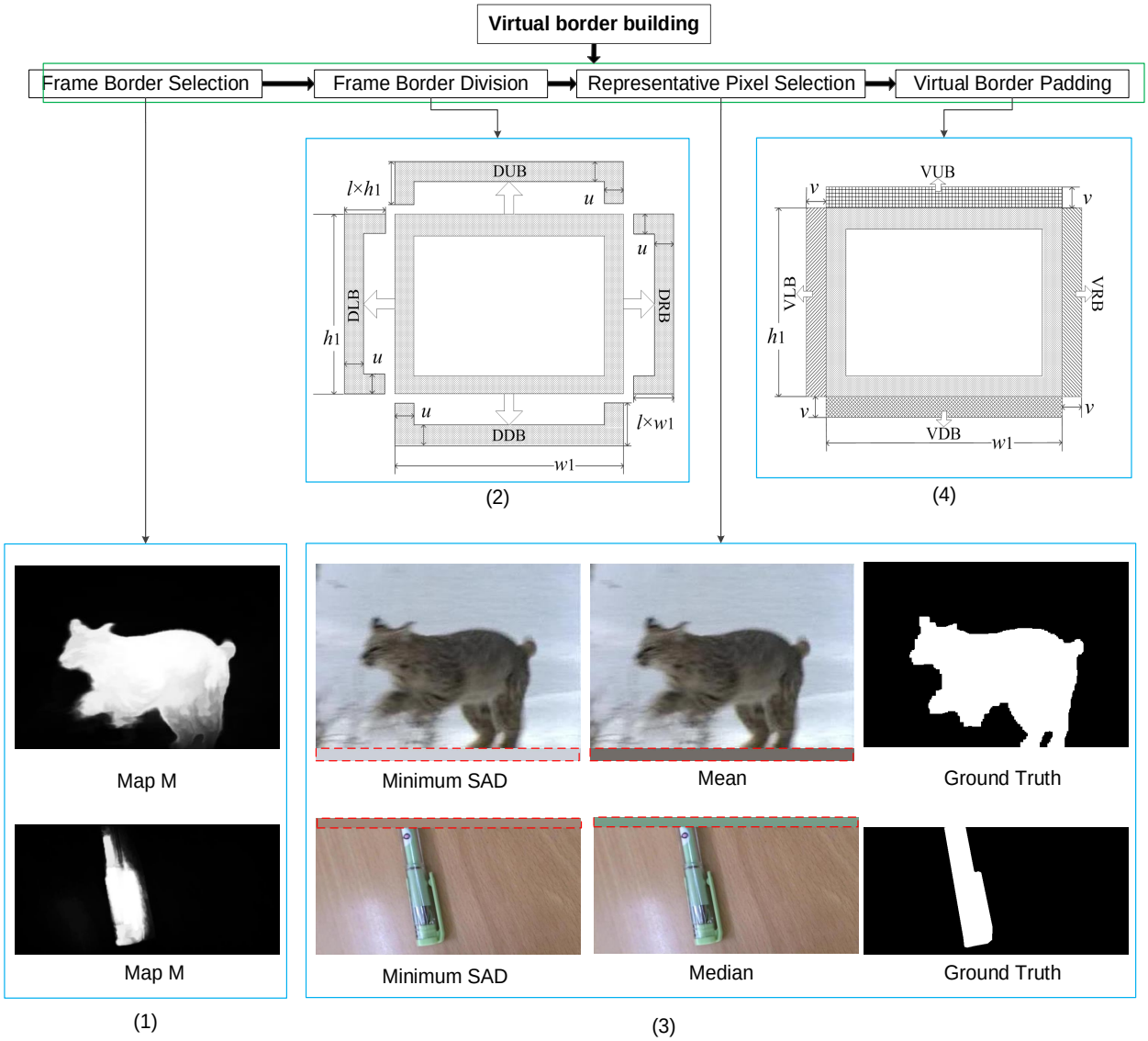


Figure 2.14: Virtual border building: (1): two examples of map M obtained by applying FastMBD on the frame; and then for each frame, the closest border to the salient region is selected to build the virtual border; (2): generating the divided border from the highlighted frame border (with width u), h_1 is the frame height, w_1 is the frame width and l is the ratio of the corresponding border length, four divided borders: the DUB, the DDB, the DLB and the DRB are shown; (3): two examples of the representative pixel selection, where “Mean” means the representative pixel is chosen using the mean value of the border’s intensities and “Median” means choosing the median value of the border’s intensities as the representative pixel, the red dotted line denotes the virtual border padded with the selected representative pixel; (4): building and padding the virtual border (with size v) with representative pixel value, four virtual borders: VUB, the VDB, the VLB and the VRB, are shown in four different textures.

shape connecting three borders is used to calculate the virtual border. The parameters u and l are selected empirically. In this chapter, u is set to 5 and l is set to 18%. Preliminary experiments showed that these values make the algorithm robust to various background complexities.

c) Representative Pixel Selection: for the generated divided border, sum of absolute differences (SAD) is computed for each pixel by summing all the absolute differences between this pixel and other pixels in the divided border:

$$\mathrm{SAD}(x) = \sum_{x' \in \mathrm{DB}} |I(x) - I(x')| \tag{2.2}$$

where $\mathrm{DB} \in \big\{$DUB, DDB, DLB and DRB$\big\}$, I is the feature channel. The pixel having the minimum SAD is selected to represent the divided border. For color images, the SAD is computed by summing the three color channels:

$$\mathrm{colorSAD}(x) = \sum_{x' \in \mathrm{DB}} \sum_{i \in \{r,g,b\}} \left| I^i(x) - I^i(x') \right| \tag{2.3}$$

We have also considered using the mean or median value of the border's intensities as the representative pixel value. Various experiments conducted on different frames have shown that the minimum SAD choice performs better than the mean and the median values in most of the cases (cf. the 1st example image in Fig.2.14 where the representative pixel is chosen from the salient object instead of the background when using the mean value of the border's intensities). The same way, choosing the median value of the border's intensities as the representative pixel value fails, which can be seen on the 2nd example image in Fig.2.14. As the minimum SAD performs better in most cases and in order to be more robust in all situations, we adopt the minimum SAD in the proposed method.

d) Virtual Border Padding: around the selected original frame border, we build the corresponding virtual border with the above representative pixel to get the with-virtual-border frame D. The VUB, the VDB, the VLB and the VRB are shown in the middle right part in Fig.2.14. Existing methods usually regard the border (with width 1) to be background and seed sizes are set to be 1. Here we set the virtual border size v to 5, which helps the proposed "virtual border building" to be applied to other distance transform based saliency detection methods.

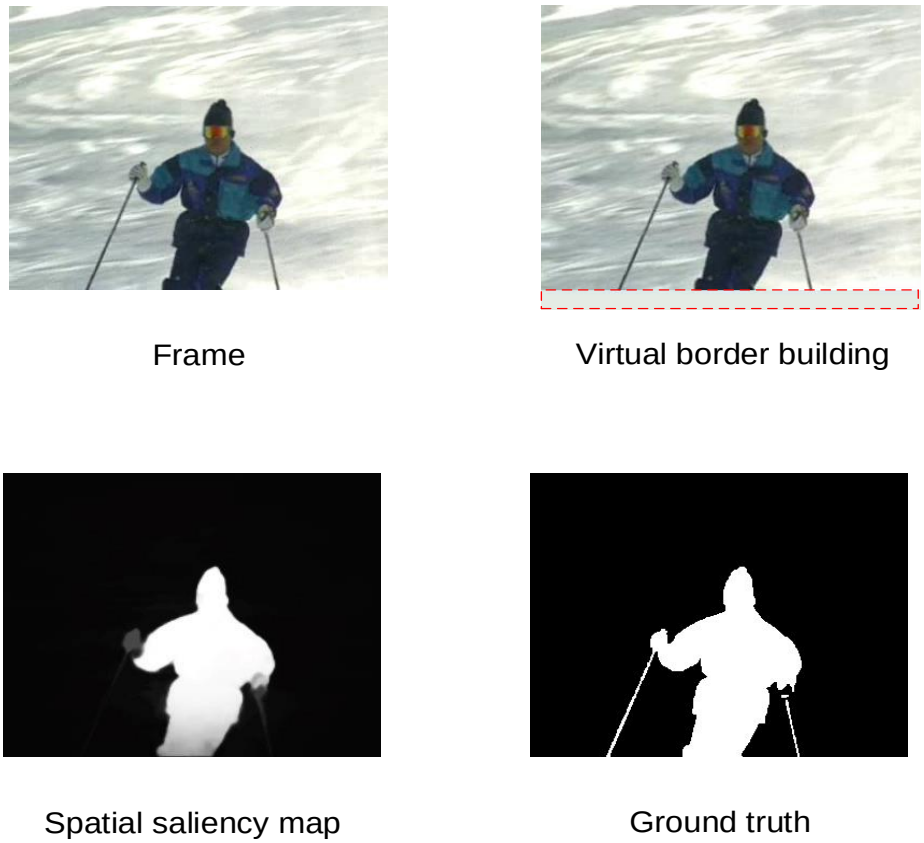


Figure 2.15: (Better viewed in color) An example of the spatial saliency detection. The red dotted line denotes the virtual border.

Saliency computation

After the "virtual border building", the spatial saliency map SSM is obtained by apply the FastMBD [103] to the with-virtual-border frame D and then remove the virtual border from the resulted map to obtain the spatial saliency map SSM. One example is given to show the process of spatial saliency detection in Fig.2.15.

2.3.2 Temporal saliency detection

In temporal saliency detection (TD), given an input video sequence, the movement information is extracted from the whole video and then the salient object is detected from this movement information. This part is related to the method we called TGFV and

published in TGFV17 [87].

Movement extraction

The optical flow vectors between pairs of successive frames are obtained using a fast optical flow method [33]. Then the optical flow vector is mapped to Munsell color system [2] to produce the color optical flow map E.

Virtual border building

Based on the background cue, the global motion is usually connected to E borders. The global motion is mainly generated by the background and camera motion. Camera motion appears in the whole color optical flow map and background motion has a high probability to be connected with E borders. Thus, E borders can reflect the global motion caused by both the background motion and the camera motion. The distance of each pixel to the border pixels of E calculated by the FastMBD [103] can indicate its temporal saliency. The larger the distance, the higher the temporal saliency value. As the same problem in the spatial saliency detection, when the salient object touches frame borders, its movement information also touches E borders. If we directly apply the FastMBD [103] on E, the salient object movement part connected to E borders is hard to be detected. Thus, we add virtual borders on E using the same procedure as described in Section 2.3.1 to obtain the with-virtual-border color optical flow map F.

Feature fusion

In our spatial saliency detection, only color and luminance features are used to detect the saliency, while edges are inherent features of the image and intrinsically salient for visual perception. Though some researches detect the salient object by considering edges, their results may be still inaccurate. Thus we propose a new Feature fusion way that fuses the spatial edge with the temporal information, considering that: 1) the salient object movement is often bigger than the background movement, thus the background and the salient object are often shown in different colors in the color optical flow map; 2) if the movements within the salient object are different, the salient object cannot be detected completely. If the spatial edges are added onto the optical flow map F, the salient object edges will be enhanced. The pixel's distance in blur edges will be

increased if the pixel belongs to the salient object or decreased if the pixel belongs to the background. Thus we performed the guided filtering. The guided filter [31] is a linear filtering process, which involves a guidance image C^1, an input image C^2 and an output image C^3. The C^3 at a pixel i is computed using the filter kernel K which is a function of C^1 but independent of C^2.

$$C^3{}_i = \sum_j K_{ij}(C^1)C^2{}_j, \tag{2.4}$$

where i and j are pixel indexes, and

$$K_{ij}(C^1) = (|\omega_k|)^{-2} \sum_{(i,j)\in\omega_k} (1 + (C^1{}_i - \mu_k)(C^1{}_j - \mu_k)({\sigma_k}^2 + \epsilon)^{-1}), \tag{2.5}$$

where ω_k is the square window centered at the pixel k in C^1, $|\omega_k|$ is the number of pixels in ω_k, ϵ is a regularization parameter, and μ_k and ${\sigma_k}^2$ are the mean and the variance of C^1 in ω_k. The main assumption of the guided filter is a local linear model between C^1 and C^3. Thus, C^3 has an edge if C^1 has an edge.

The proposed method uses with-virtual-border frame D as the guidance image and with-virtual-border color optical flow map F as the input image to get the filtered image G as Eq (2.6),

$$G_i = \sum_j |\omega_k|^{-2} \sum_{(i,j)\in\omega_k} (1 + (D_i - \mu_k)(D_j - \mu_k)({\sigma_k}^2 + \epsilon)^{-1})F_j, \tag{2.6}$$

where i and j are pixel indexes, ω_k is the square window centered at the pixel k in D_i, μ_k and ${\sigma_k}^2$ are the mean and the variance of D_i in ω_k. ϵ is set to be 10^{-6}. $|\omega_k|$ is decided by the frame size. Large frame needs large ω_k. We use 20×20 for Fukuchi and FBMS datasets, and use 60×60 for VOS dataset since VOS has larger average frame size than that of Fukuchi and FBMS [27, 49].

Saliency computation

The FastMBD [103] is applied on the filtered image G and then the virtual border region is removed to obtain the temporal saliency map TSM. One example is given to show the process of the temporal saliency detection in Fig.2.16.

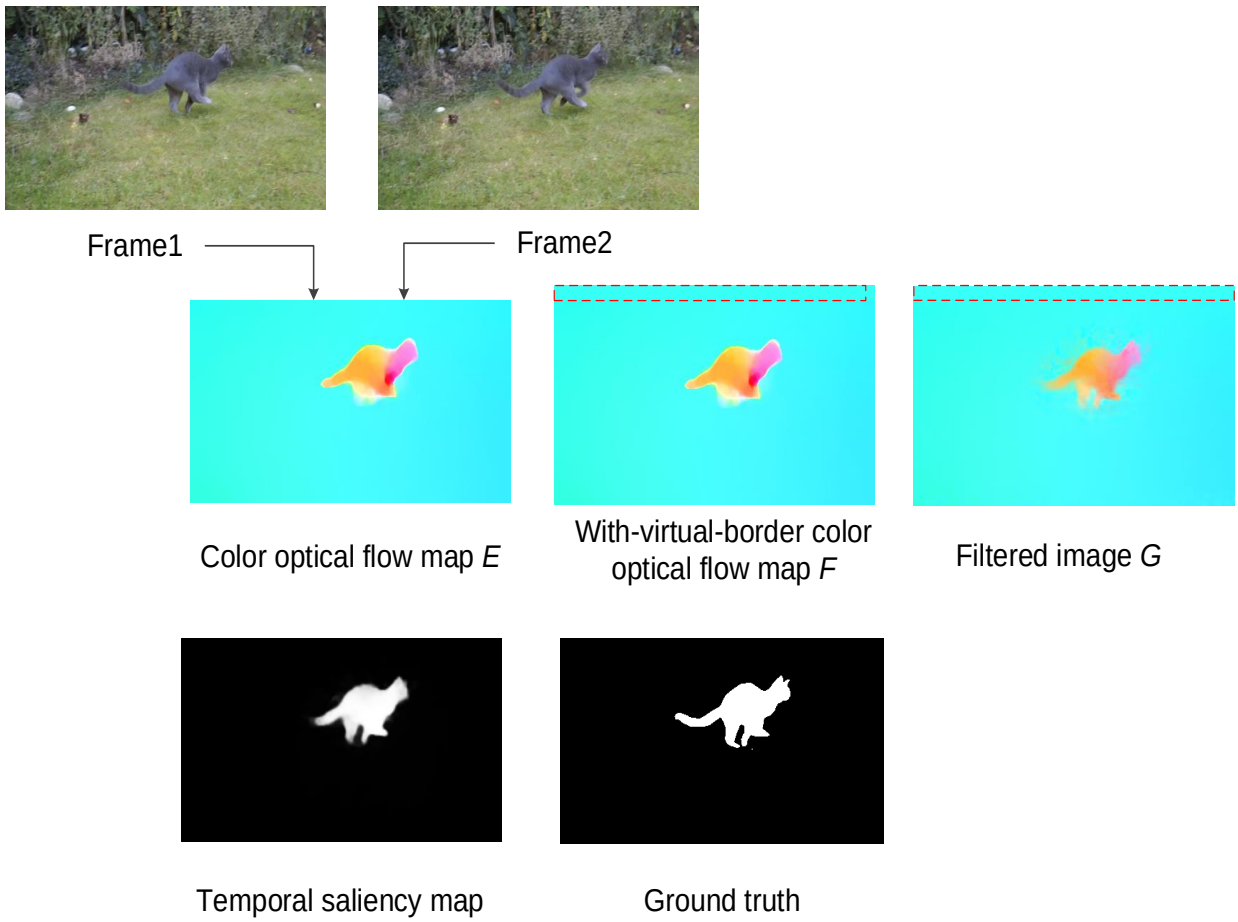


Figure 2.16: (Better viewed in color) An example of the temporal saliency detection: from two successive frames, the optical flow vector is extracted and mapped to be the color optical flow map E. The virtual border is built on map E to generate with-virtual-border color optical flow map F. The red dotted line denotes the virtual border. After guided filtering, the filtered image G is generated to produce the temporal saliency map. Ground truth is provided for comparison.

2.3.3 Spatial and temporal saliency maps fusion

Given the spatial saliency map SSM and the temporal saliency map TSM, the fusion is made to obtain STSM by four steps:

- SSM and TSM are firstly fused as Eq (2.7), where $\mathrm{ratio}_1 = \mathrm{mu}_T/(\mathrm{mu}_S + \mathrm{mu}_T)$, $\mathrm{ratio}_2 = 1 - \mathrm{ratio}_1$.

$$\mathrm{STSM} = \mathrm{ratio}_1 \times \mathrm{SSM} + \mathrm{ratio}_2 \times \mathrm{TSM} \tag{2.7}$$

  where mu_S and mu_T are respectively the mean entropies of all the spatial saliency maps and all the temporal saliency maps for a video sequence (with $\varkappa$ the number of frames) as Eq (2.8).

$$\begin{aligned} \mathrm{mu}_S &= \sum_{j=1}^{\varkappa}(-\sum_{j'=1}^{255}(\mathrm{Prob}_{j'}^{S^j} \times \log(\mathrm{Prob}_{j'}^{S^j})))/\varkappa \\ \mathrm{mu}_T &= \sum_{j=1}^{\varkappa}(-\sum_{j'=1}^{255}(\mathrm{Prob}_{j'}^{T^j} \times \log(\mathrm{Prob}_{j'}^{T^j})))/\varkappa \end{aligned} \tag{2.8}$$

  where $\mathrm{Prob}_{j'}^{S^j}$ and $\mathrm{Prob}_{j'}^{T^j}$ are respectively the normalized histogram of j^{th} spatial saliency map and j^{th} temporal saliency map: $\mathrm{Prob}_{j'} = \mathrm{num}_{j'}/(h_1 \times w_1)$, $\mathrm{num}_{j'}$ is the number of pixel (equal to j') in saliency map. Here, the idea is that mu_i ($i = S, T$) are used to decide the confidence of SSM and TSM. The disorder degree of saliency map reflects the difficulty degree to detect the salient objects. If mu_i ($i \in \{S, T\}$) is larger, the saliency detection in this domain is worser.

- STSM is optimized using Eq (2.9)

$$\mathrm{STSM} = \mathrm{SSM} \qquad if \quad \mathrm{mu}_S < \mathrm{mu}_T \tag{2.9}$$

  The frame is often more complex than the color optical flow map, which results in that the disorder degree of SSM is usually larger than that of TSM. If mu_S is smaller than mu_T, it means it is difficult to detect the salient object in TSM. Thus, SSM has a high confidence.

- STSM is optimized using Eq (2.10)

$$\mathrm{STSM} = \mathrm{SSM} \qquad if \quad \sigma_S > \sigma_T \tag{2.10}$$

σ_S and σ_T are respectively the standard deviations of non-zero regions in two grayscale images H_S and H_T, which are generated by the following steps: firstly, converting frame α from RGB to HSI color space, then eliminating the hue and saturation information while retaining the luminance to get the grayscale images α'; secondly, using a threshold δ to neglect the pixels with low saliency value from the images SSM and TSM as in Eq (2.11)

$$H_{S_{ij}} = \begin{cases} 0 & if \quad \mathrm{SSM}_{ij} < \delta \\ \alpha'_{ij} & otherwise \end{cases} \qquad H_{T_{ij}} = \begin{cases} 0 & if \quad \mathrm{TSM}_{ij} < \delta \\ \alpha'_{ij} & otherwise \end{cases} \tag{2.11}$$

where i and j are pixel indexes in the images. The appearance of the wrongly detected background is mostly different from the salient object in the grayscale image, which results in that H_i ($i \in \{\mathrm{S,T}\}$) contains more luminance values and thus σ_i ($i \in \{\mathrm{S,T}\}$) is smaller. If σ_S is bigger than σ_T, it means SSM has a high confidence.

- Low saliency value (lower than δ) in SSM is decreased to 0.1 times.

  The pixels with low saliency value in saliency map are unimportant for visual saliency but have a large influence in computing the detection confidence. Thus, δ is used to decrease their impact and set to 70 in this chapter.

2.4 Experiments and analyses

In this section, the performance of the proposed method VBGF is assessed and discussed. The performance of each component of the VBGF is shown to demonstrate our contributions. The VBGF's performance is then compared with nine state-of-the-art traditional SOD methods. Finally, the run-time complexity is compared.

In order to fully evaluate the effectiveness and robustness of the proposed method against the state-of-the-art methods of the same category, two popular related datasets FBMS and Fukuchi are used.

Nine state-of-the-art saliency models are tested: MST16 [80], FastMBD15 [103], AMC13 [36], TGFV17 [87], SGSP16 [54], RWR15 [39], GF15 [89], SAG15 [88], FD17 [14] on Fukuchi and FBMS dataset. For all the methods, the experimental results are obtained using the source codes or saliency results provided by the authors.

2.4.1 Contributions of each proposed component to the performance

The contributions are shown by analyzing the performance of each component.

Contribution of the proposed Virtual Border Building

The method (based on the "background prior") may miss the salient object connected to the image borders and the proposed virtual border aims to improve this problem. Since MST16 [80], FastMBD15 [103] and AMC13 [36] detect the salient object in image domain based on the "background prior", we compare the proposed spatial saliency map with them by using the Fukuchi dataset, in which many salient objects connected to the frame border. Quantitative performance can be found in Fig.2.17. The proposed spatial saliency detection has a better performance since it can detect salient objects more completely.

Contribution of the proposed Feature fusion

The proposed Feature fusion employs the guided filter to fuse the spatial edges with the information in temporal domain. We compare the performance of the proposed temporal saliency map with guided filtering and without guided filtering. In the Fukuchi dataset the salient object motion is small, and in the FBMS, the global motion varies largely. These two different datasets are both used. Quantitative performance can be found in Fig.2.18 and Fig.2.19. We can see that fusing the spatial salient object edges to the temporal information by using guided filtering can improve the detection accuracy. It helps to optimize the salient object edges and remove the background part from the saliency region.

Contribution of the proposed Map fusion

Our proposed method first generates spatial saliency map (cf. Section 2.3.1), then generates the temporal saliency map (cf. Section 2.3.2), finally generates the spatio-temporal saliency map (cf. Section 2.3.3). Therefore, we separately test the performance of each proposed saliency map, then compared quantitative results can be found in Fig.2.20 and Fig.2.21. For the Fukuchi dataset, the salient object motion is slow while the salient object and the background are in high contrast. Compared

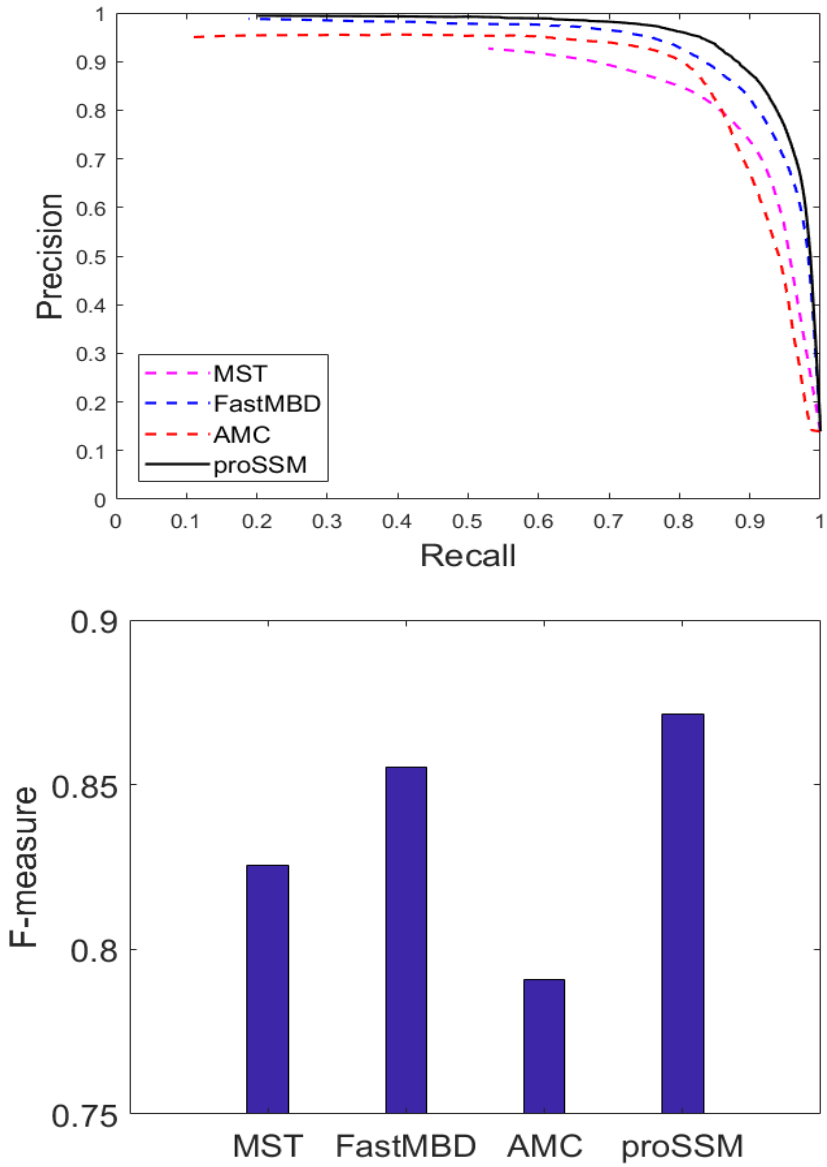
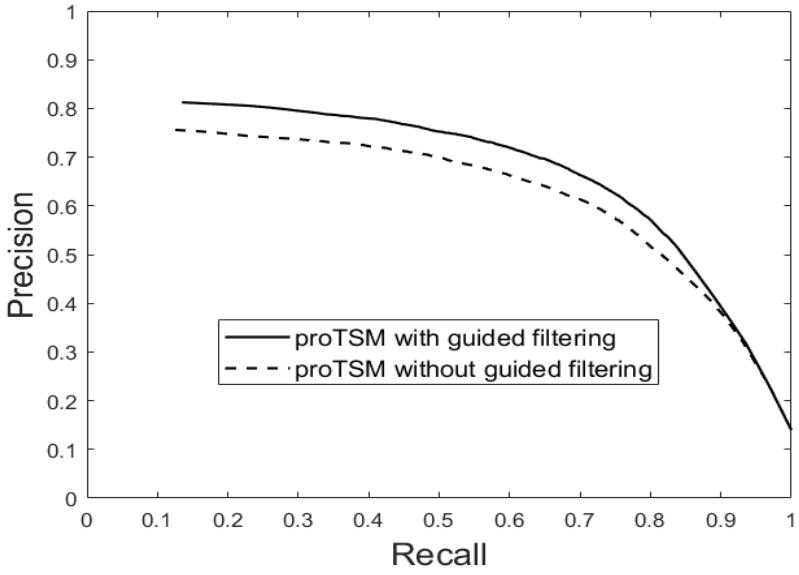
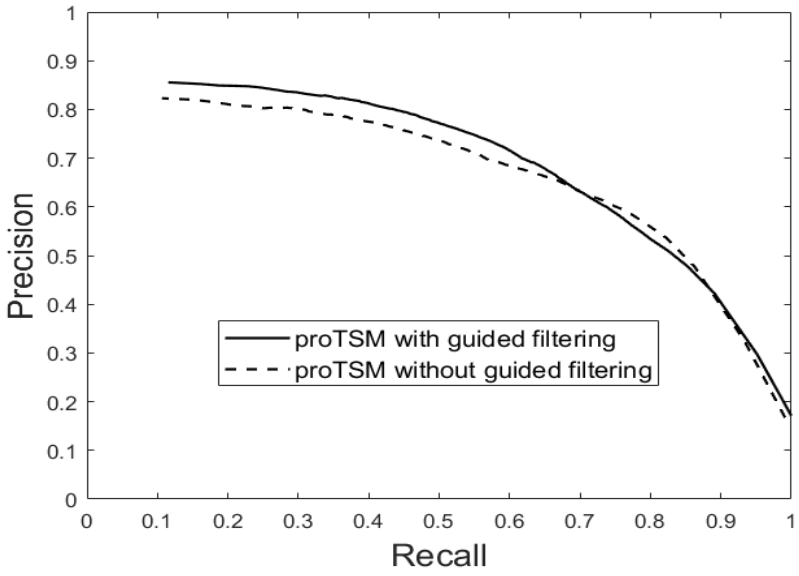


Figure 2.17: (Better viewed in color) Quantitative comparisons between our proSSM (proposed spatial saliency map) and three image SOD models over the Fukuchi dataset. Some state-of-the-art methods, including: MST16 [80], FastMBD15 [103] and AMC13 [36]. The left parts show the P-R curves, the right parts shows the F-measure scores ↑.



P-R curves over Fukuchi



P-R curves over FBMS

Figure 2.18: P-R curves of proTSM (proposed temporal saliency map) with guided filtering and without guided filtering over the Fukuchi dataset and the FBMS dataset.

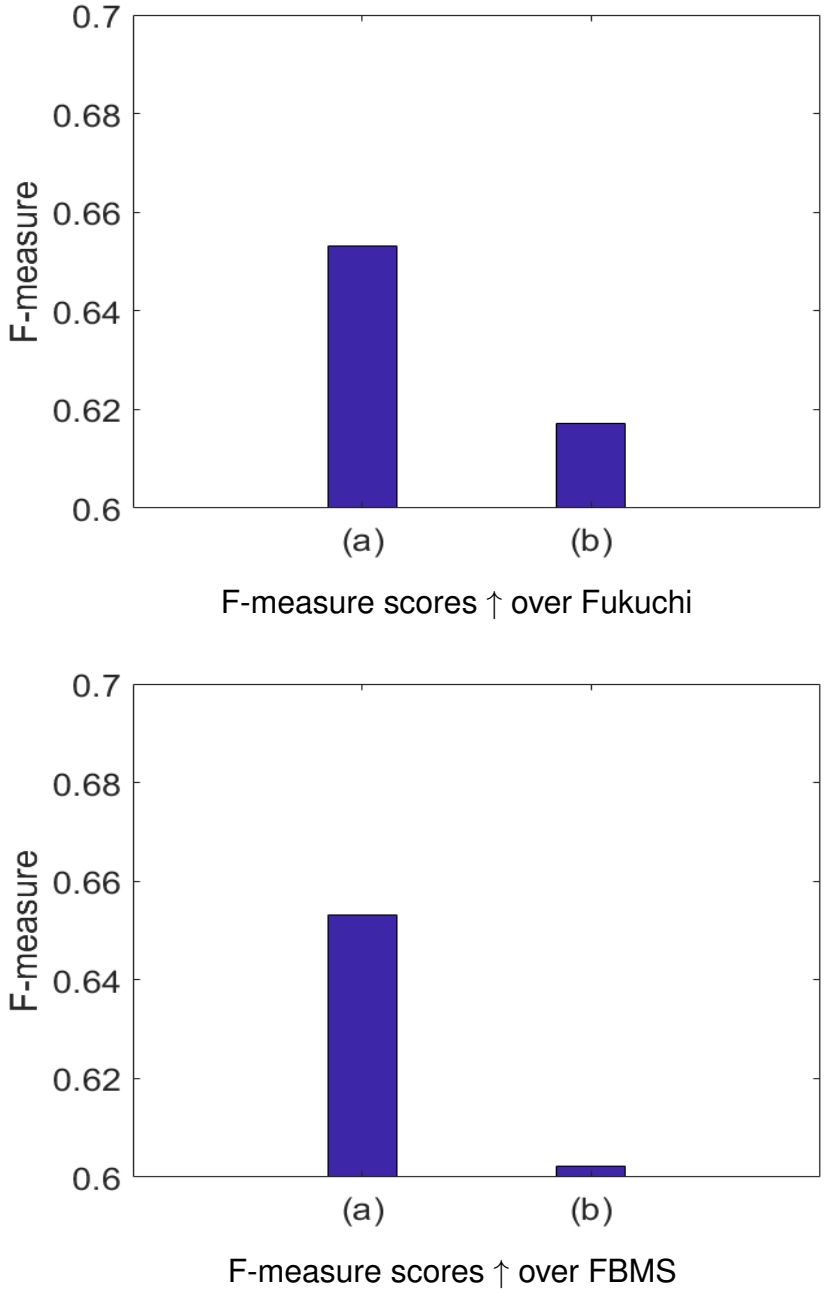
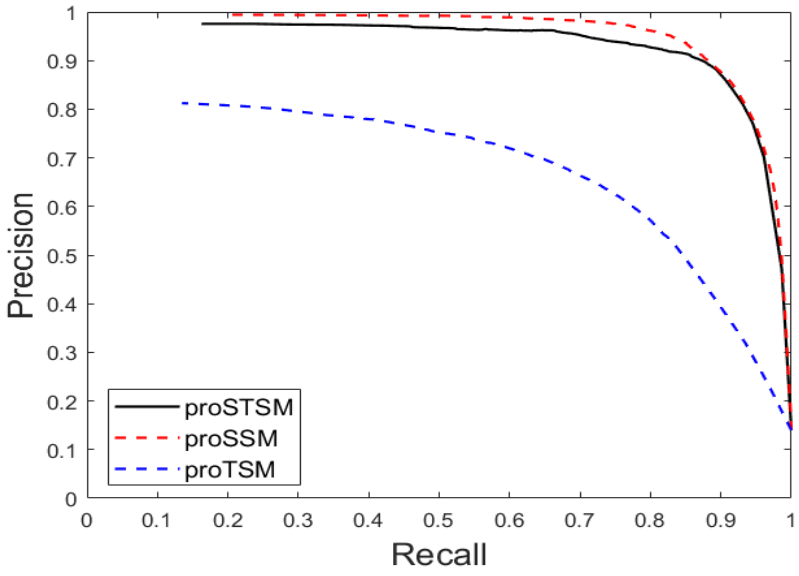
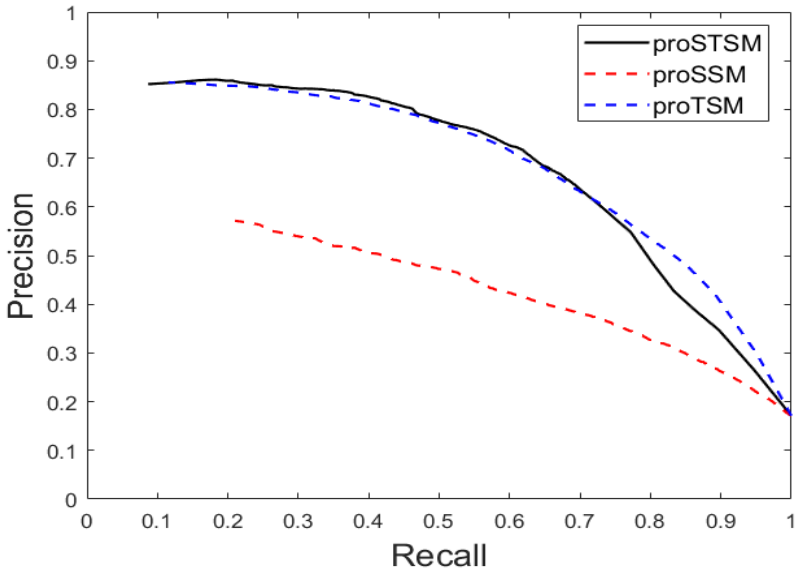


Figure 2.19: F-measure scores of the proposed temporal saliency map: (a) with guided filtering and (b) without guided filtering over the Fukuchi dataset and the FBMS dataset.

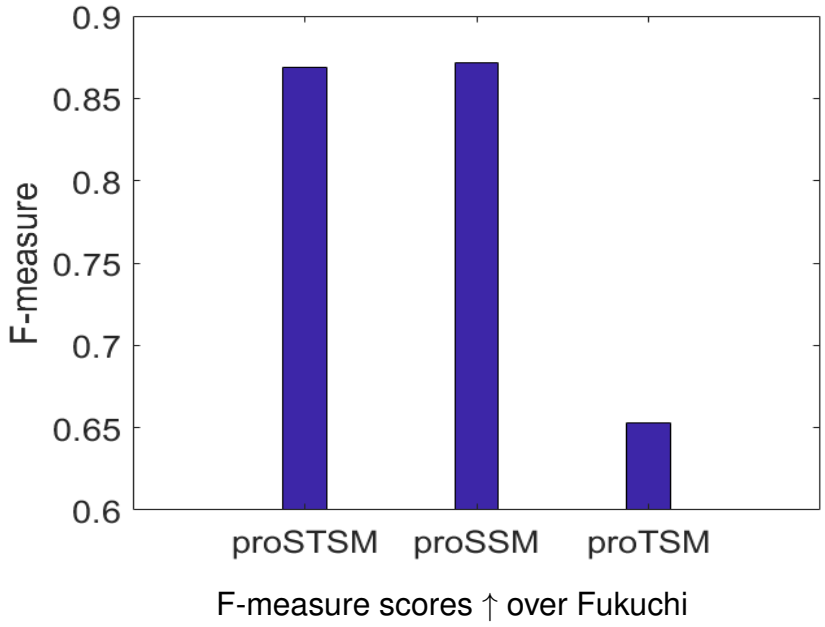


P-R curves over Fukuchi

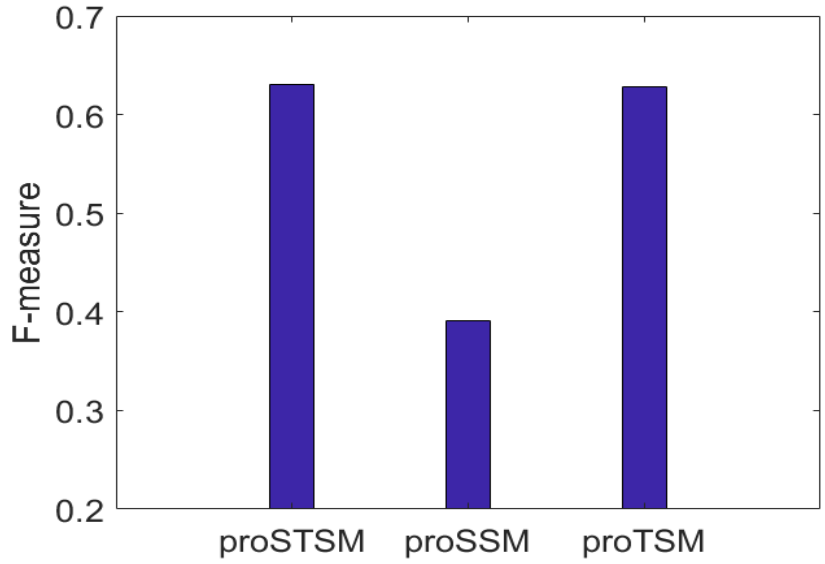


P-R curves over FBMS

Figure 2.20: (Better viewed in color) P-R curves of proSSM, proTSM and proSTSM over the Fukuchi dataset and FBMS dataset. proSSM: proposed spatial saliency map; proTSM: proposed temporal saliency map; proSTSM: proposed spatio-temporal saliency map.



F-measure scores ↑ over Fukuchi



F-measure scores ↑ over FBMS

Figure 2.21: F-measure scores of proSSM, proTSM and proSTSM over the Fukuchi dataset and the FBMS dataset. proSSM: proposed spatial saliency map; proTSM: proposed temporal saliency map; proSTSM: proposed spatio-temporal saliency map.

with the spatial saliency map, the detected temporal saliency has a lower confidence. The proposed fusion can still get a good performance by retaining the spatial saliency map while neglecting the temporal detection influence. For the FBMS dataset, the low contrast and the complex background in the spatial domain make the spatial saliency detection inaccurate. Though the global motion is intricate, the temporal saliency map is still better than the spatial saliency map. The proposed fusion method takes advantages of results from both domains and gives a higher overall performance.

2.4.2 Comparison of the proposed method with state-of-the-art methods

Quantitative comparison with video SOD models

We compare our proposed method (VBGF, also called as proSTSM) with several video SOD models with the Fukuchi dataset and the FBMS dataset respectively.

For the Fukuchi dataset, six compared models are: TGFV17 [87], SGSP16 [54], RWR15 [39], GF15 [89], SAG15 [88], FD17 [14]. The P-R curves, F-measure and MAE values are drawn in Fig.2.22, from which we can see that the proposed method has the best P-R curve, the highest F-measure and the smallest MAE values. The detailed MAE and F-measure scores over four video sequences are shown in Table.2.1 and the proposed method achieves the best performance. In the Fukuchi dataset, the contrast between the salient object and the background is large and the salient object movement is slow. Spatial saliency detection thus can already provide a high confidence, while the wrong detections in the temporal domain may influence the final saliency map. Compared with methods TGFV17 [87], SGSP16 [54], RWR15 [39], GF15 [89], SA15 [88], and FD17 [14], the proposed fusion method can better select higher confidence spatial saliency information from two domains.

For the FBMS dataset, five compared models are TGFV17 [87], SGSP16 [54], RWR15 [39], GF15 [89], SAG15 [88]. Fig.2.23 reports the P-R curves, F-measure and MAE values. We can see that our proposed method performs the best, while all the methods get lower performances on this dataset since it is the most challenging one. Five videos with difficult cases (the salient object is similar to the background or the background is complex) are selected and the detailed corresponding MAE and F-measure scores are shown in Table.2.2, in which the proposed method is always the best method. In the FBMS dataset, on one hand, the global motion exists in many

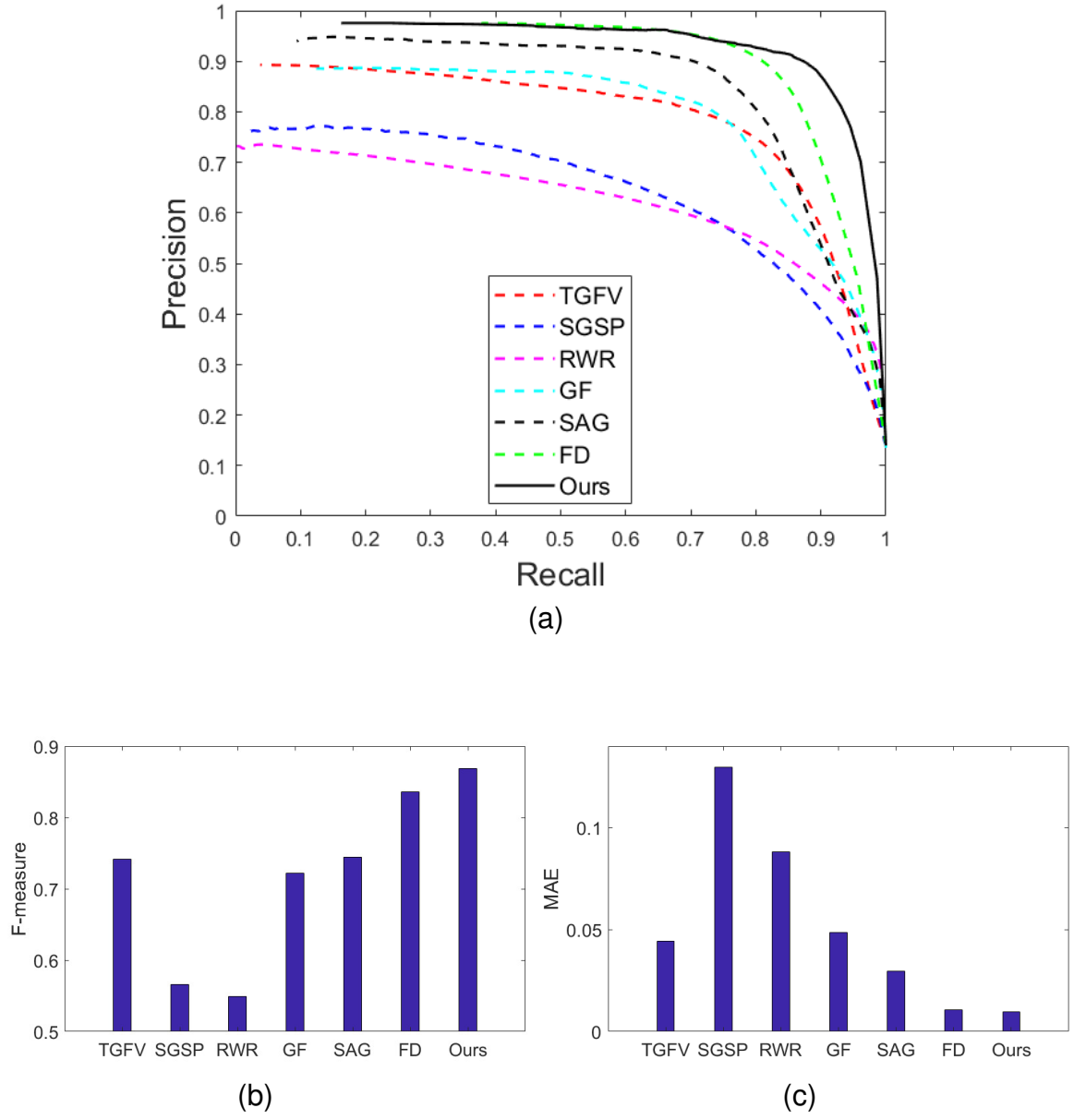


Figure 2.22: (Better viewed in color) Quantitative comparisons between our method (VBGF) and six video SOD models over the Fukuchi dataset. (a) show the P-R curves, (b) shows the F-measure scores ↑ and (c) shows MAE scores ↓. Some state-of-the-art methods, including: TGFV17 [87], SGSP16 [54], RWR15 [39], GF15 [89], SAG15 [88], FD17 [14]

.

Table 2.1: A table comparing the our method (VBGF) and six video SOD models in MAE ↓ and F-measure ↑ scores over 4 video sequences chosen from the Fukuchi dataset. The Bold number indicates the best result.

Method	MAE scores ↓			
	AN119T	DO01_013	DO01_055	DO02_001
TGFV17 [87]	0.0119	0.0084	0.0462	0.0324
SGSP16 [54]	0.0772	0.0675	0.0996	0.1463
RWR15 [39]	0.0692	0.0773	0.052	0.0826
GF15 [89]	0.0312	0.0306	0.0334	0.0378
SAG15 [88]	0.0264	0.0247	0. 026	0.0162
FD17 [14]	0.0062	0.0086	0. 0165	0.0113
VBGF	**0.0027**	**0.0052**	**0.0053**	**0.0014**
Method	F-measure scores ↑			
	AN119T	DO01_013	DO01_055	DO02_001
TGFV17 [87]	0.9069	0.704	0.7228	0.808
SGSP16 [54]	0.7318	0.6343	0.5411	0.5925
RWR15 [39]	0.4878	0.5379	0.6533	0.6182
GF15 [89]	0.8659	0.6842	0.7417	0.8292
SAG15 [88]	0.8432	0.5486	0.7393	0.8348
FD17 [14]	0.9449	0.685	0.7852	0.8656
VBGF	**0.9516**	**0.801**	**0.8051**	**0.9322**

sequences and is with high complexity which make the temporal detection more difficult. On the other hand, the salient object appearance is similar to that of the background and the background is complex which makes the spatial detection more difficult. Among the compared methods (TGFV17 [87], SGSP16 [54], RWR15 [39], GF15 [89] and SAG15 [88]), TGFV17 [87] gets a better result since it puts emphasize on the temporal saliency detection. However, compared with TGFV17 [87], the proposed method leverage the spatial saliency and fuses them in a more efficient way to obtain better result.

Subjective comparison with video SOD models

To evaluate the overall performances and disparities between our method and the state-of-the-art methods, we also show a subjective comparison in Fig.2.24 and Fig.2.25. We can see that RWR15 [39] tends to detect salient object edges rather than the whole salient object. Methods : MST16 [80], FastMBD15 [103], AMC13 [36], TGFV17

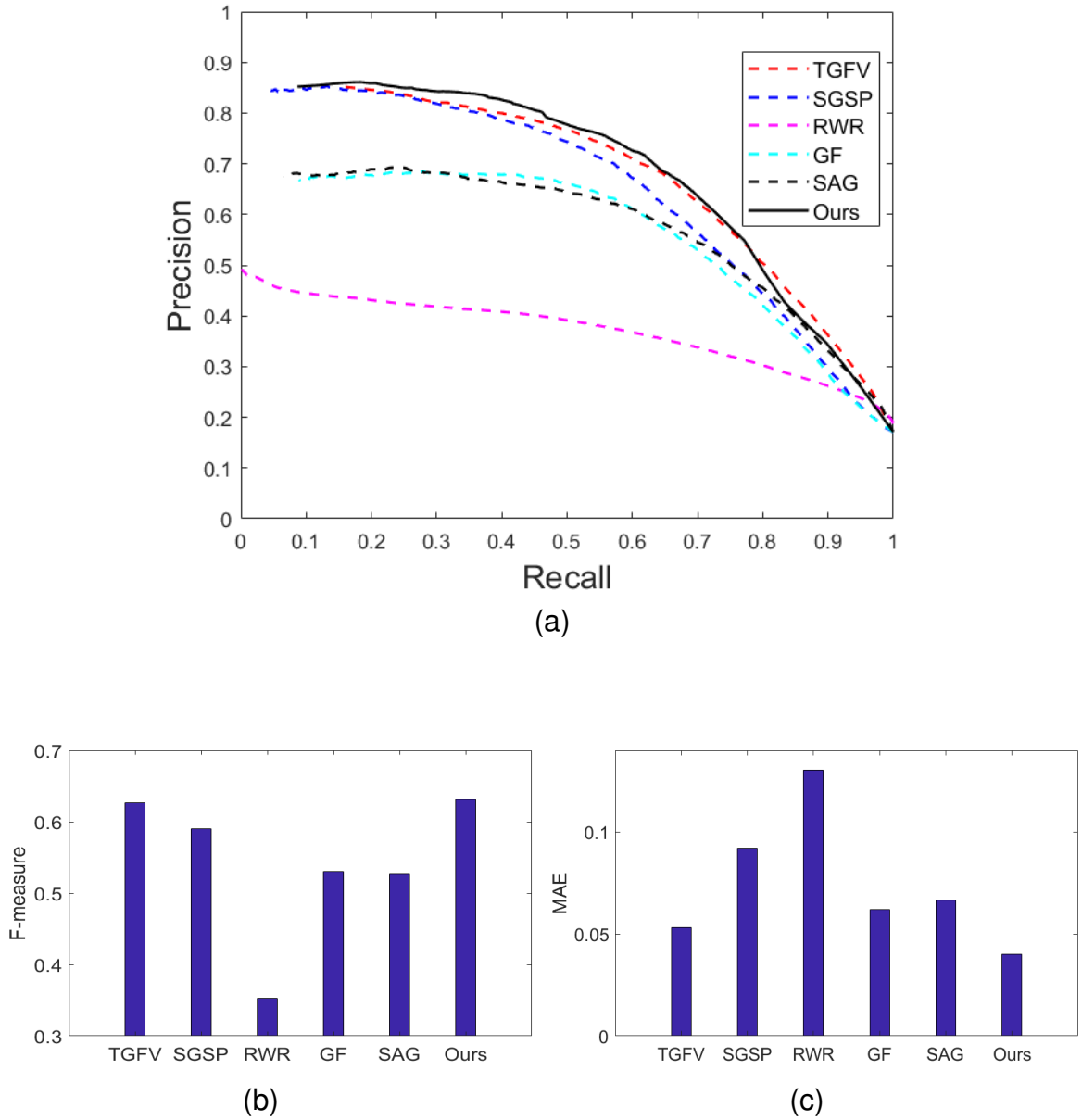


Figure 2.23: (Better viewed in color) Quantitative comparisons between our method (VBGF) and five video SOD models over the FBMS dataset. (a) show the P-R curves, (b) shows the F-measure scores ↑ and (c) shows the MAE scores ↓. Some state-of-the-art methods, including: TGFV17 [87], SGSP16 [54], RWR15 [39], GF15 [89] and SAG15 [88].

Table 2.2: A table comparing the our method (VBGF) and five video SOD models in MAE ↓ and F-measure scores ↑ over 5 video sequences chosen from the FBMS dataset.

Method	MAE scores ↓				
	Cars5	Cars10	Cats03	Horses04	Horses05
TGFV17 [87]	0.0205	0.0248	0.0536	0.0454	0.0363
SGSP16 [54]	0.0708	0.0599	0.1089	0.0964	0.0877
RWR15 [39]	0.1905	0.1485	0.1471	0.1175	0.0968
GF15 [89]	0.0438	0.0388	0.1148	0.1049	0.0598
SAG15 [88]	0.0486	0.034	0. 0941	0.1427	0.0689
VBGF	**0.0161**	**0.0218**	**0.0103**	**0.0243**	**0.0215**
Method	F-measure scores ↑				
	Cars5	Cars10	Cats03	Horses04	Horses05
TGFV17 [87]	0.751	0.6494	0.6573	0.7021	0.6018
SGSP16 [54]	0.6359	0.6595	0.6558	0.6476	0.6105
RWR15 [39]	0.3485	0.4056	0.2219	0.3389	0.3666
GF15 [89]	0.5877	0.6339	0.2762	0.6415	6067
SAG15 [88]	0.4964	0.584	0.3532	0.3797	0.6495
VBGF	**0.7712**	**0.7281**	**0.7184**	**0.7294**	**0.6593**

[87], SGSP16 [54], GF15 [89], SAG15 [88] can detect salient object region located in the frame center but not the salient part close to frame borders. Especially, when the salient object exhibits clearly distinctive color features from the background, e.g. (e) and (g), the salient object connected to borders is detected with low saliency in the above methods. However, the proposed algorithm yields good performances on these cases. In (b) and (d), it's difficult to distinguish the edge between the salient object and the background for the spatial-only methods MST16 [80], FastMBD15 [103] and AMC13 [36]. While among video saliency models, the method TGFV17 [87] and the proposed method can detect the salient object with less spatial influence and more accurate edges than TGFV17 [87], SGSP16 [54], RWR15 [39], GF15 [89] and SAG15 [88]. For multiple salient objects with complex background, e.g. (i), (j) and (k), TGFV17 [87] and the proposed method can detect almost all multiple salient objects, but the proposed method has better edges. By visually comparing on this figure, we can see that the proposed method can detect the salient object more completely and more accurately.

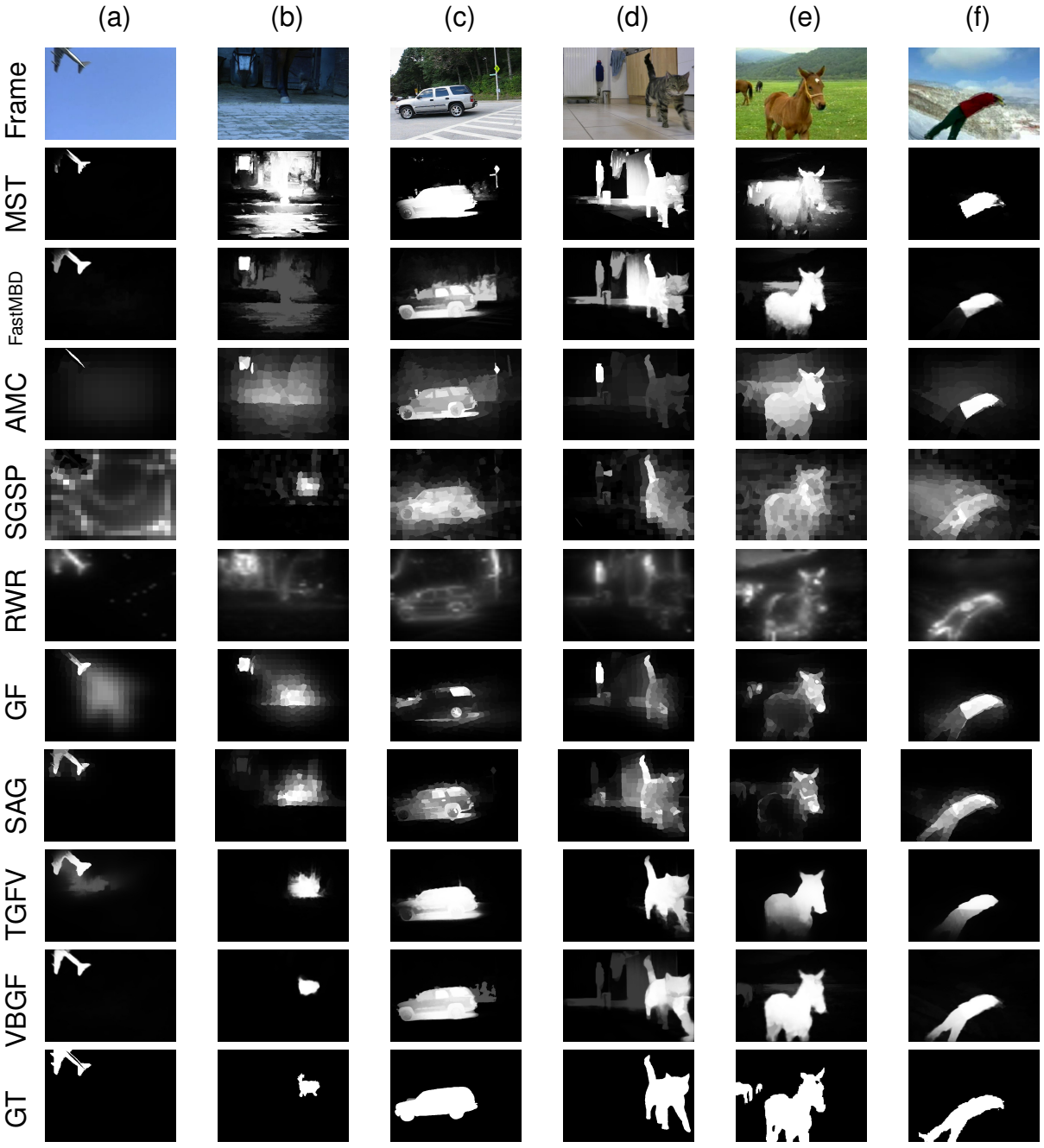


Figure 2.24: Comparison of the saliency maps (1). (a)-(f) are 6 different video sequences. Some state-of-the-art methods, including: MST16 [80], FastMBD15 [103], AMC13 [36], TGFV17 [87], SGSP16 [54], RWR15 [39], GF15 [89], SAG15 [88].

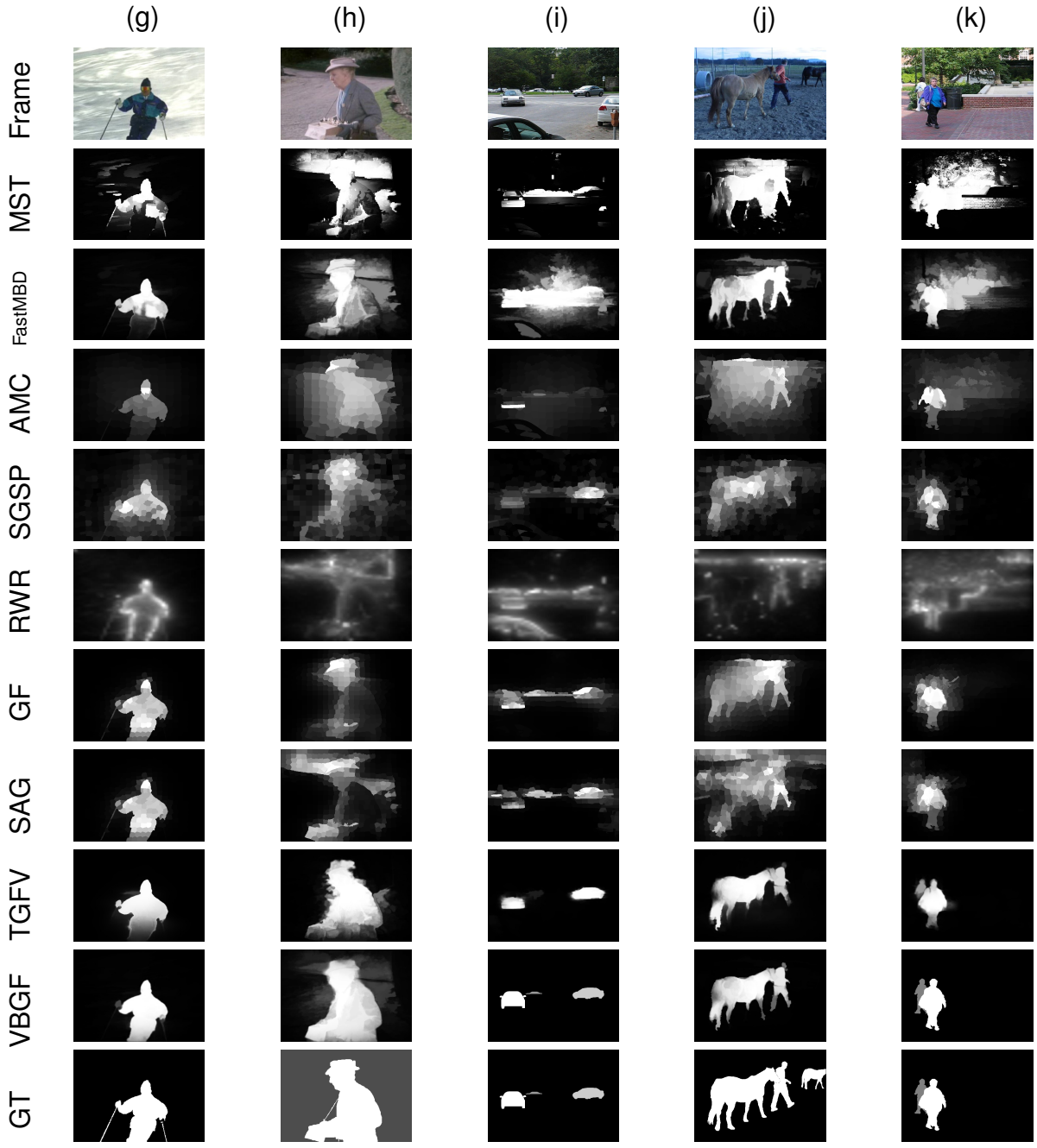


Figure 2.25: Comparison of the saliency maps (2). (g)-(k) are 5 different video sequences. Some state-of-the-art methods, including: MST16 [80], FastMBD15 [103], AMC13 [36], TGFV17 [87], SGSP16 [54], RWR15 [39], GF15 [89], SAG15 [88].

2.4.3 Computation time comparison

A PC with Intel Core i7 4910 2.9GHz CPU and 16GB RAM is used for testing the speed of traditional methods. For different models (except the model FD17 [14] with the unpublished code), the average run-time is listed in Table 2.3. Video models have higher computation costs than the other 3 image models since the optical flow estimation is usually time consuming. Our proposed model is the fastest video detection model, and the average run-time per frame of each processing stage can be found in Table 2.4 in detail.

Table 2.3: Average run time (per frame) of our proposed method (VBGF) and the compared models (MST16 [80], FastMBD15 [103], AMC13 [36], TGFV17 [87], SGSP16 [54], RWR15 [39], GF15 [89], SAG15 [88]).

Image_based	MST	FastMBD	AMC	-	-	-
Time(s)↓	0.200	0.018	0.153	-	-	-
Video_based	SGSP	RWR	GF	SAG	FD	VBGF
Time(s)↓	15.37	14.25	13.50	15.38	33.17	3.56

Table 2.4: Average run time (per frame) of each component in the proposed models.

Component	VBGF	
	Time(s)↓	Ratio(%)
virtual border building	0.50	14.04
saliency detection	0.07	1.97
optical flow computation	2.80	78.65
feature fusion(guided filtering)	0.07	1.97
map fusion	0.12	3.37
total	3.56	100

2.5 Conclusion

In this chapter, a novel video SOD method (the VBGF) is proposed. Using virtual border concept has helped to address the problem of distance transform employed for saliency

computation in previous approaches. The guided filter-based Feature fusion and the Map fusion are efficiently used for fusing spatial and temporal information together by applying appropriate balance. When tested on various video databases, the proposed approach yields satisfactory performance and even outperforms the state-of-the-art methods.

The virtual border can be used as an optimization operation for salient object detection methods that are based on background prior. The guided filter-based Feature fusion helps to remove background regions for moving object detection and segmentation. The Map fusion provides a new way to combine various individual saliency maps into a more robust one.

CHAPTER 3

OVERVIEW OF DEEP-LEARNING METHODS FOR SALIENT OBJECT DETECTION IN VIDEOS

In this chapter, Section 3.1 introduces existing surveys and benchmarks related to salient object detection in videos. Section 3.2 gives a classification of state-of-the-art methods, and details the frameworks of some representative methods. Section 3.3 gives comparative experimental results of these representative methods. The assessment of their performance generalities are discussed. Section 3.4 shows an extension of the proposed VBGF to integrate deep-learning technique. Section 3.5 concludes the chapter.

3.1 Summary of existing surveys and benchmarks

Recently, several researchers tend to solve the problems of SOD in videos using deep-learning methods, which largely improves the performance both for the accuracy and the efficiency. However, there is few related survey. Table 3.1 lists the most relevant works, from which we can see that former works mainly focus on traditional methods. Among the recent works related to deep-learning methods, the survey presented in [29] is only for images; and the benchmark [49] only compares deep-learning methods proposed for images with traditional methods proposed for videos. The survey of existing deep-learning methods for SOD in videos is less explored.

This chapter has two main motivations:

- Deep-learning for video SOD is an important topic and still have a large space to explore so it is interesting to have a general idea about the existing methods which may pave the way for future works.

Table 3.1: Comparison of the existing survey/benchmark for SOD

	Year	Benchmark	Survey	Traditional	Deep-learning	Video	Image
[65]	2014	×	✓	✓	×	×	✓
[6]	2014	×	✓	✓	×	×	✓
[5]	2015	✓	×	✓	×	×	✓
[29]	2018	×	✓	×	✓	×	✓
[49]	2018	✓	×	✓	✓	✓	✓

- Deep-learning methods can achieve high performance, but it heavily relies on the training datasets. Thus it is necessary to test the generality of the state-of-the-art methods through experimental comparison on different public datasets.

3.2 Introduction to state-of-the-arts methods

Deep-learning based methods for video SOD, focusing on learning the high-level features [19], gain great research interests, and some methods [17, 41, 43, 44, 50, 74, 76, 90, 92] are proposed. However, there still lack sufficient methods for comprehensive analysis. Inspired by [49], the inherently correlated tasks like video foreground object segmentation [9, 18, 78], moving object segmentation [77] and image SOD [52, 57, 85] are considered for analysis and comparison in this work.

In this section, we firstly classify the existing methods in 3.2.1, and secondly introduce in more details some representative methods of which the source codes are provided by authors in 3.2.2.

3.2.1 Classification based on the deep representations generation

According to whether the used neural network has to be trained, existing methods can be classified into two categories: 1) **off-the-shelf deep features** and 2) **multi-stage/end-to-end trained**. The used deep representations of the first category are directly extracted from existing deep networks. Thus, this is a simple way to directly use these deep representations for further researches [17, 43]. In the second category, methods usually get more efficient deep representations through their own training phase, where the inputs-outputs relationship is learned by deep architectures. The

model trained in multiple stages is with intermediate supervision to ones trained end-to-end. According to their utilization degree of the labeled datasets, the models can be further divided into supervised and weakly-supervised models.

Supervised models need training datasets with pixel-wise annotations. According to the domain of the learned deep representation, supervised methods [9, 18, 41, 44, 50, 52, 57, 74, 77, 78, 90, 92] can be classified into 1) spatial [9, 50, 52, 57]; 2) temporal [77]; 3) or spatio-temporal [18, 41, 44, 74, 78, 90, 92]. Due to the fact that current datasets have limited manually labeled ground truth, some methods, e.g. [90], propose to generate simulated video data using synthesizing methods. Different from supervised methods, **weakly-supervised models** train the network without requiring all training datasets to have corresponding pixel-level annotations. Some models learn to detect the salient object from spatial domain with image-level annotations, based on the assumption that image-level tags can provide the classes of the dominant objects which can be regarded as the salient foregrounds, e.g. [85]. Sometimes, a small number of manually labeled data and a huge amount of weakly labeled data are used together. For example, in [76], one seventh of the frames in a video is manually labeled data and the rest is weakly labeled. Three existing SOD methods are used to generate the weakly labeled data, and their proposed network is trained using both manually and weakly labeled data. Then the weakly labeled data is updated using their proposed network, as well as the three existing SOD methods. Fig. 3.1 shows the classification of the deep-learning based SOD methods.

3.2.2 Description of salient object detection frameworks

This section gives detailed introduction of 11 representative methods, which the source codes or saliency results are provided by the authors. Among them, Chen *et al.* [17] propose a off-the-shelf deep features based model, and methods in [9, 18, 50, 52, 57, 76–78, 85, 90] are multi-stage/end-to-end trained models. Methods in [9, 18, 50, 52, 57, 77, 78, 90] are supervised models and with those in [76, 85] are weakly-supervised models.

Firstly, the global framework for each method is described and then the deep network designed in each method is analyzed.

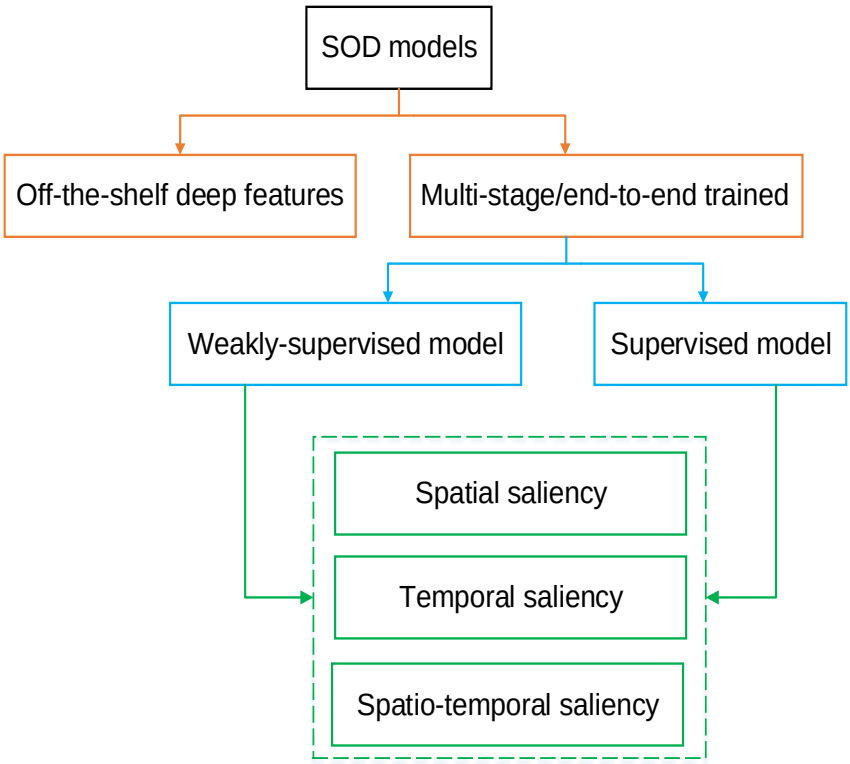


Figure 3.1: Methods classification according to the deep representations generation

Analysis of the frameworks of representative methods

As a matter of convenience, 11 methods are denoted as SCOMd [17], NRF [50], DHSNet [52], OSVOS [9], NLDF [57], LMP [77], SFCN [90], SegFlow [18], LVO [78], WSS [85], SCNN [76].

According to the involved tasks, these 11 frameworks can be divided into two categories: multi-task [18, 85] and single-task [9, 17, 50, 52, 57, 76–78, 90].

The **multi-task framework** not only predicts the salient objects, but also evaluates other tasks. It exploits the connections between the SOD task and other highly related tasks (such as classification, contour detection, optical flow and boundary detection), and then improves the SOD performance by making use of the deep representation from these tasks. Specifically, Wang *et al.* [85] propose a weakly-supervised network which has two subnetworks: one is designed for classification and the other is designed for SOD. Firstly, using image-level tags as the ground truth, detection stream is

jointly trained with the classification subnetwork for classification prediction. Secondly, the saliency prediction of the detection subnetwork is used as the ground truth for fine-tuning the detection subnetwork. Both subnetworks share convolutional layers firstly and then are separated on the top of the shared layers, as shown in Fig. 3.2 (a). Cheng *et al.* [18] propose a supervised network which also consists of two subnetworks: the segmentation subnetwork and the flow subnetwork. A bi-directional feature propagation is built between these two networks as shown in Fig. 3.2 (b), and an iterative training is used for optimizing the segmentation task. The OSVOS proposes two fully convolutional networks (FCNs) with the same architecture. The first FCN is used as a foreground branch and the second FCN is employed as a contour branch. The output of the first FCN is optimized by combining with that from the second FCN, as shown in Fig. 3.2 (c). The NLDF, an end-to-end trained network, adds the boundary loss term to design extra constraints to saliency prediction.

The **single-task framework** is designed just for the SOD task. Among them, SFCN and SCNN propose two fully convolutional networks (FCN) with the same architecture in their frameworks. From Fig. 3.3 (a), Wang *et al.* [90] use the first FCN for spatial saliency detection with the input of each frame, and use the other FCN for spatio-temporal saliency detection with the input of adjacent frame pairs and the detected spatial saliency results. The detected spatial saliency results is denoted as SFCNs. From Fig. 3.3 (b), Tang *et al.* [76] firstly employ one FCN to get a spatial prior map, secondly generate temporal prior map from optical flow fields, thirdly combine these two prior maps to be a spatio-temporal prior map which guides the second FCN to generate the spatio-temporal saliency map. At last, the output saliency map is optimized by a CRF model.

The SCOMd, NRF, LMP and DHSNet models only adopt one network in their single-task frameworks. In SCOMd, the authors employ a pretrained network and uses the deep spatial features instead of the handcrafted features, to define a new motion energy for SOD in video. In NRF, the authors firstly obtain the initial salient object and background estimation with their complementary convolutional neural network, and then construct a neighborhood reversible flow to propagate salient object and background along the most reliable inter-frame correspondences. The NRF is summarized in Fig. 3.4 (a). DHSNet and LMP, as in Fig. 3.4 (b), are end-to-end training networks without any other processing. In LMP, the authors detect motion patterns in videos with designed motion pattern network. While, In LVO, the authors firstly use the network

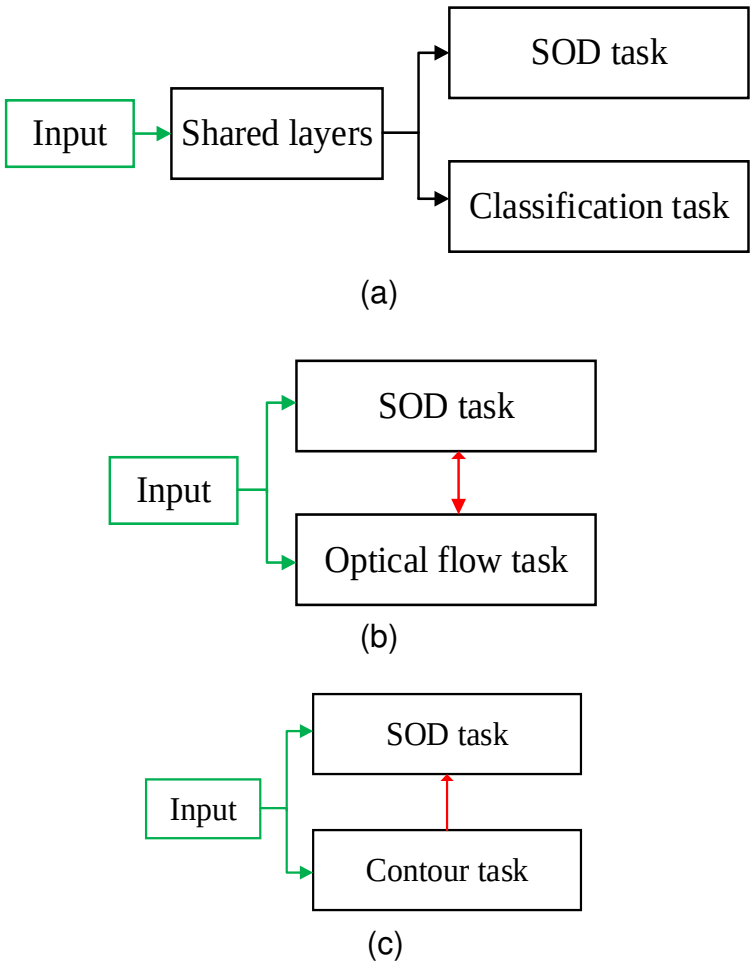


Figure 3.2: Multi-tasks models: (a) WSS, (b) SegFlow and (c) OSVOS

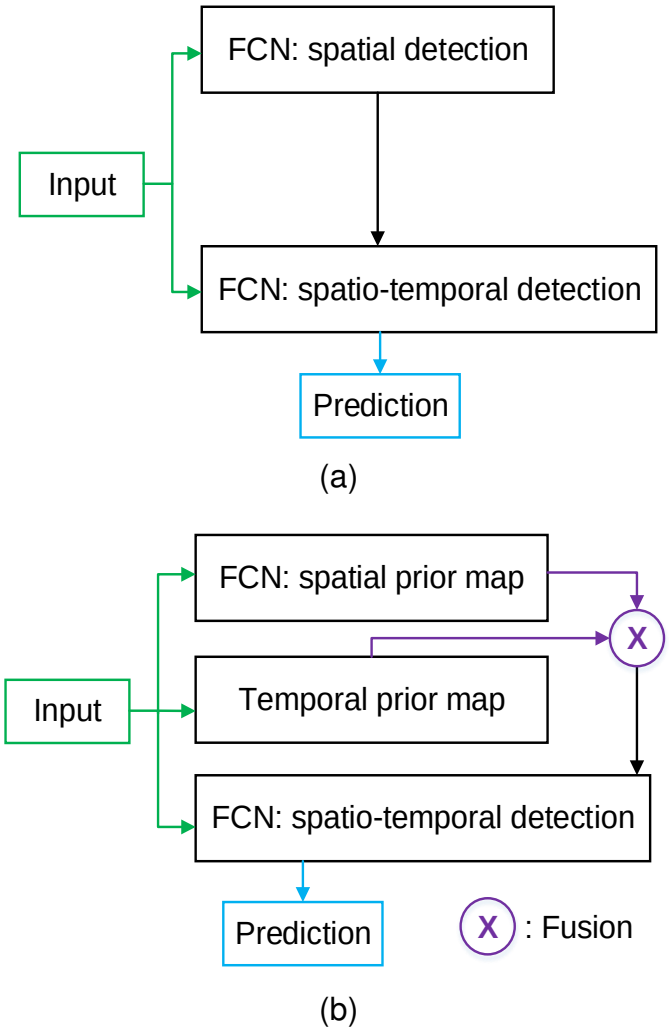


Figure 3.3: Single-task models: (a) SFCN and (b) SCNN

proposed in [16] to extract deep spatial features in the appearance stream, and then adopt the network proposed in [77] to detect motion patterns in the motion stream, and thirdly build a visual memory module which inputs the concatenation of appearance and motion streams to get the prediction. The LVO is shown in Fig. 3.4 (c).

Analysis of the deep networks of representative methods

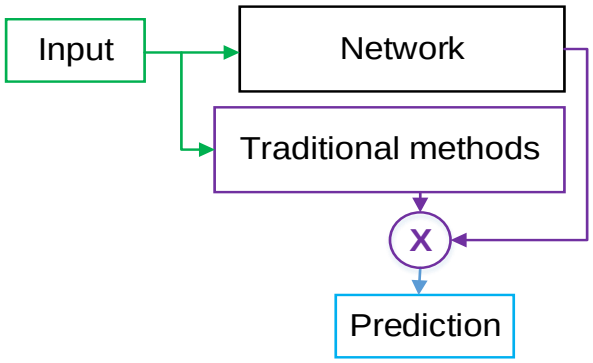
In this part, we analyze the networks designed in the representative methods.

A typical network for SOD is usually an encoder-decoder network, and hierarchical features are generated layer by layer, as shown in Fig. 3.5.

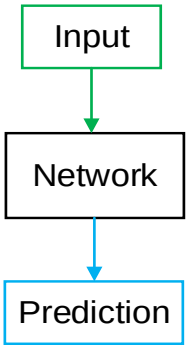
The methods, e.g. [9, 50, 52, 57, 77, 85], use skip connections [9, 18, 52, 57, 77, 85] or "à trous" pyramid pooling (ASPP) [50] to employ multi-scale feature maps for prediction. These networks are illustrated in Fig. 3.6. Specifically, Tokmakov *et al.* [77] add skip connections from the encoder features to the mirror decoder features, which benefits the decoder features with finer details. Cheng *et al.* [18], Wang *et al.* [90] and Wang *et al.* [85] mainly use feature maps from 3rd to 5th layers of the backbone, while Tang *et al.* [76] considers responses from 4th and 5th layers for predicting the final output. Luo *et al.* [57] add multiple skip connections to fully employ the deep information. Liu *et al.* [52] add skip connections between mirror layers, but with multiple predictions. Four predictions in Fig. 3.6 (d) are used in the training step. And only the last one is used to generate the final saliency result. Caelles *et al.* [9] add skip connections from the low-level layer to the high-level layer. Feature maps obtained from each layer are fused into a single output. Li *et al.* [50] use three parallel modules with ASPP to capture the multi-scale information. The outputs (Prediction1 and Prediction2 in Fig. 3.6 (f)) are both used to generate the saliency result.

Table 3.2 summarizes the used backbone and training datasets for each mentioned representative method.

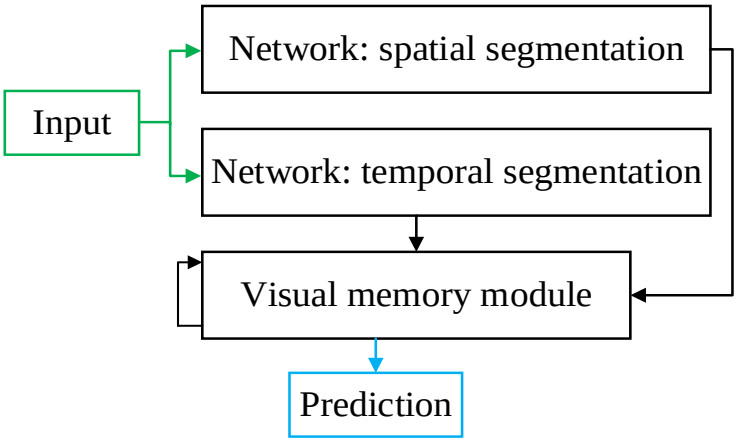
Networks for SOD often built the encoder network based on a backbone (i.e. an existing trained model with published weights). Image classification networks (e.g. VGG [73] and ResNet [32]) are commonly used as backbones. These networks [32, 73] are trained on large-scale image datasets and have a strong ability to learn both low-level and high-level features. Note that various networks are proposed based on VGGNet or ResNet for dense prediction. FlowNetS [21] is only used for estimating the optical flow and the baseline in [18] to obtain the temporal feature.



(a)



(b)



(c)

Figure 3.4: Single-task models: (a) NRF, (b) DHSNet and LMP, (c) LVO.

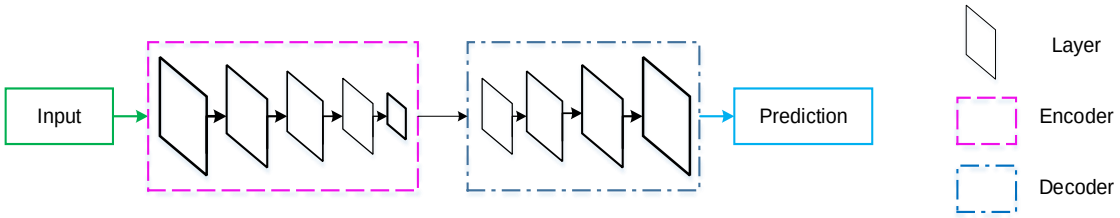
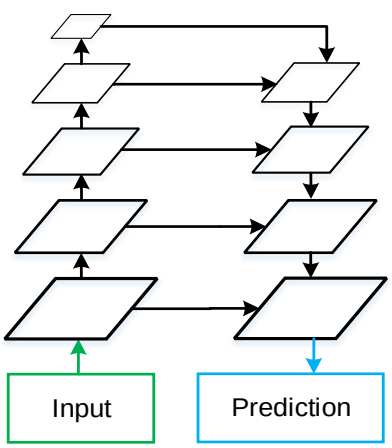
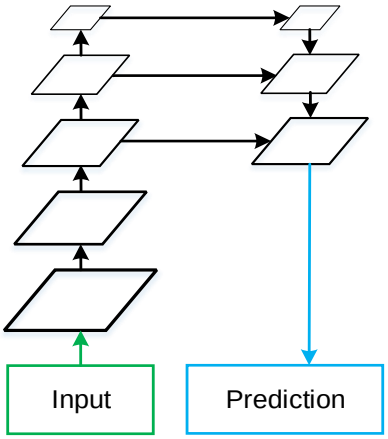


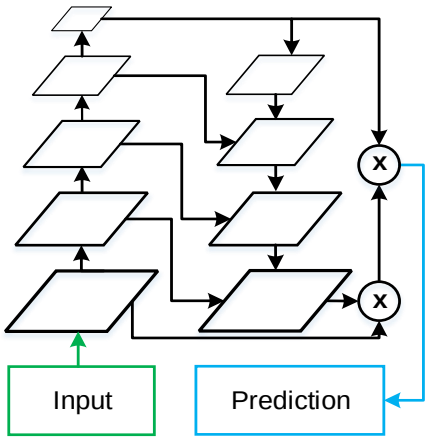
Figure 3.5: Encoder-decoder network



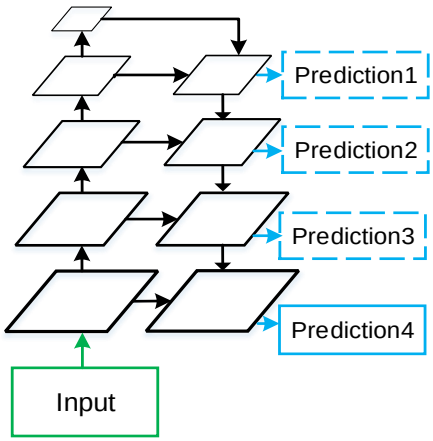
(a)



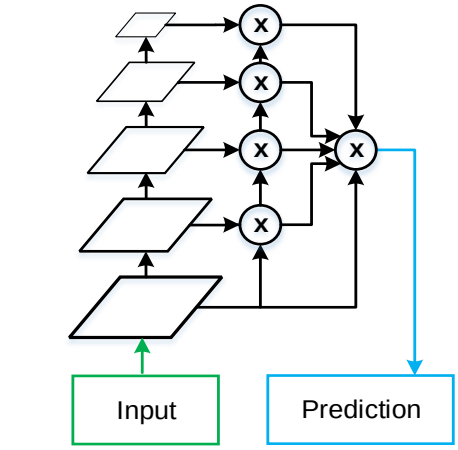
(b)



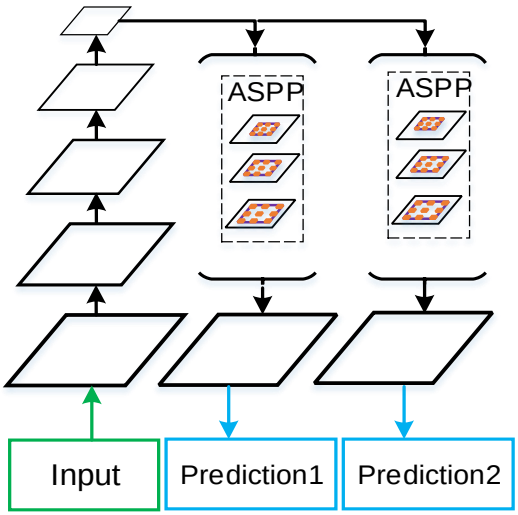
(c)



(d)



(e)



(f)

Figure 3.6: Networks: (a) [77], (b) [18, 85], (c) [57], (d) [52], (e) [9], (f) [50].

Table 3.2: Backbone and Training datasets ("x" indicates that the method is not based on any backbone or the method is off-the-shelf deep features based)

Methods	Backbone	Training datasets
SCOMd[17]	VGG16	x
NRF[50]	VGG16	HKU-IS,MSRA10K,CSSD,DUT-OMRON
DHSNet[52]	VGG16	MSRA10K,DUT-OMRON
OSVOS[9]	VGG16	DAVIS 2016-train,PASCAL-Context
NLDF[57]	VGG16	MSRA-B
LMP[77]	x	FlyingThings3D
SFCN[90]	VGG16	MSRA10K,SegTrackV2,DUT-OMRON,FBMS-training
SegFlow[18]	ResNet101,FlowNetS	DAVIS 2016-train,MPI Sintel[7],KITTI,Scene Flow
LVO[78]	VGG16	DAVIS 2016-train
WSS[85]	VGG16	DUTS
SCNN[76]	VGG16	MSRA10K,SegTrackV2,FBMS-training

Various training datasets are used for networks to learn deep representations: Image SOD datasets (e.g. MSRA-B, MSRA10K [53], DUT-OMRON [101], HKU-IS [48] and CSSD [99]) are used in most methods, e.g. [50, 52, 57, 76, 90]; image object segmentation datasets (e.g. DUTS [85]) are used in [85]; video object segmentation datasets (e.g. SegTrackV2 [46], DAVIS 2016-train) are used in methods [9, 18, 76, 78, 90]; contour datasets (e.g. PASCAL-Context [62]) are used in [9]; moving object segmentation datasets (e.g. FBMS-training [8]) are used in methods [76, 90]; optical flow datasets (FlyingThings3D [60]) are used in [77]; and datasets (MPI Sintel [7], KITTI [35], Scene Flow [63]) are used in [18]. Besides, some methods generate new datasets from existing datasets: Wang *et al.* [90] create synthesized video dataset due to the limitation of video SOD datasets, and Tokmakov *et al.* [78] create training sequences which simulate cases where the object stops moving.

During the training phase, a network learns all the parameters via minimizing errors between the result and the ground truth. A loss function is used to compute this error. The "cross entropy" is commonly used for SOD [18, 50, 52, 57, 78]. Given the generated SM and GT, the cross entropy loss *P* is given by Eq (3.1).

$$\mathrm{P} = -\sum_{i=1}^{h1\times w1} (g_i \log s_i + (1-g_i)\log(1-s_i)) \tag{3.1}$$

where h_1 is the frame height, w_1 is the frame width, $g_i \in \mathrm{GT}$ and $s_i \in \mathrm{SM}$. Since the numbers of salient and non-salient pixels are not balanced, the "balanced cross

entropy", given by Eq (3.2), is more commonly used for SOD [9, 76, 90].

$$\mathrm{P} = -\sum_{i=1}^{h1\times w1}((1-R)\times g_i \log s_i + R\times(1-g_i)\log(1-s_i)) \tag{3.2}$$

where R is the ratio of the number of salient pixels in GT over that of all pixels in GT.

Besides, motivated by the successful application of boundary Intersection over Union (IOU) loss in medical image segmentation [61], Luo *et al.* [57] add a boundary IOU loss, given by Eq (3.3), for SOD.

$$\mathrm{IOUloss} = 1 - \frac{2\left|C_{\mathrm{GT}} \bigcap C_{\mathrm{SM}}\right|}{|C_{\mathrm{GT}}| + |C_{\mathrm{SM}}|} \tag{3.3}$$

where C_{GT} and C_{SM} are contours pixels of GT and SM respectively, which are obtained using the magnitude of Sobel operator followed by a tanh activation. In order to prevent learning high responses at all locations, Wang *et al.* [85] apply sparse regularization on the generated saliency map to reduce background noise during pre-training phases.

3.3 Experimental evaluation

In order to assess the generality of the state-of-the-art methods, large-scale datasets (including FBMS, VOS-E, VOS-N, VOS, DAVIS 2016-val and DAVIS-2017-val) are used to evaluate the above mentioned 11 methods: five metrics (including MAE, Recall, Precision, F-measure and P-R curve) are used to evaluate saliency methods (SCOMd, SFCN, SFCNs, DHSNet, NLDF, WSS and SCNN) and four metrics (including MAE, Recall, Precision and F-measure) are used to evaluate segmentation methods (LMP, LVO, SegFlow, NRF and OSVOS).

For methods SCOMd and SCNN, applied to FBMS and DAVIS 2016-val datasets, the results are those reported by the authors. For methods DHSNet, NLDF, NRF, OSVOS, SFCNs and WSS, applied to all datasets, the results are generated using the provided source codes (OSVOS dose not contain the boundary snapping branch and WSS does not contain conditional random field (CRF) processing). When the authors give their results, we just report these results even if they provide their code.

Note that LMP firstly detects the motion pattern with the MP-Net, then uses the traditional spatial objectness and the CRF to refine the temporal results. The LVO also

applies a CRF as a post-processing on the detection network output. In order to explore their deep performances, we just use their network outputs, denoted as LMPd and LVOd for our comparison. For their network inputs, the optical flow vector is generated by the method proposed by Tripathi *et al.* [79].

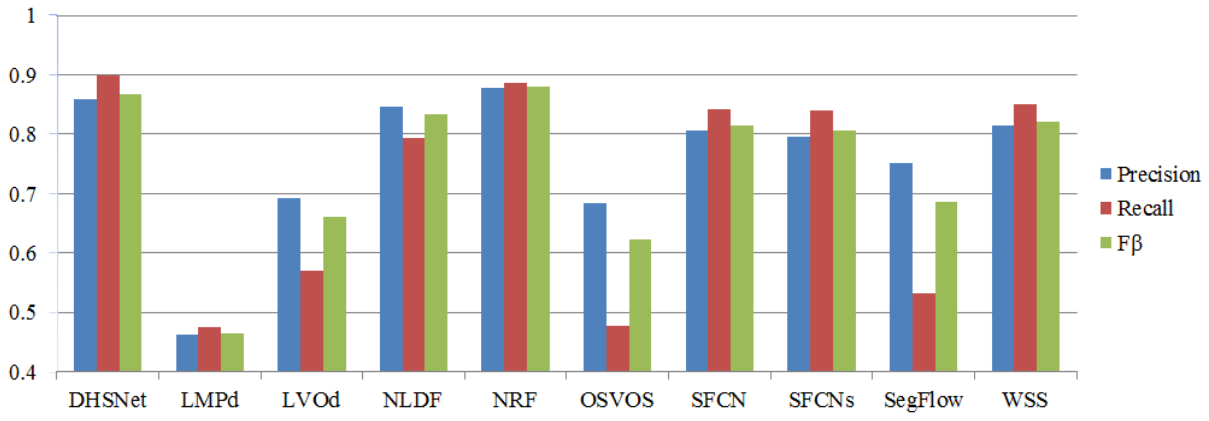
3.3.1 Detailed performance on each dataset

Performance on the VOS-E dataset

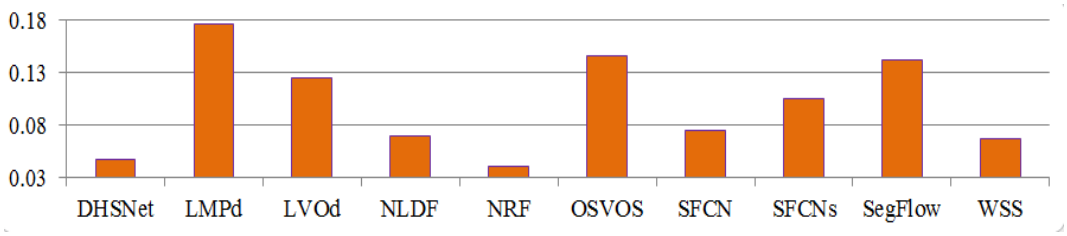
Fig. 3.7 shows the performance on the VOS-E dataset (with slow camera motion). DHSNet, NLDF, NRF, SFCN, SFCNs and WSS methods, based on backbone networks, all get high Precision, high Recall and high F-measure scores. DHSNet also gets the best P-R curve, and NRF gets the best MAE value. Most of these methods only detect the salient object from spatial domain, which shows that spatial saliency detection has a good performance for SOD on video dataset with slow camera motions.

Performance on the FBMS dataset

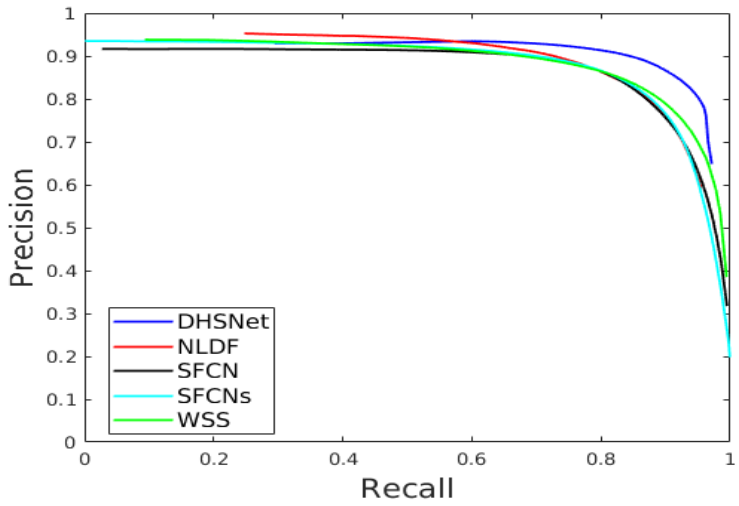
Fig. 3.8 presents the performances on the FBMS dataset which puts emphasis on the moving object. SCNN gets the best Recall score, and SCOMd gets the best Precision score, and LVOd gets the best F-measure score, and SegFlow gets the best MAE value. They not only detect the salient object from spatial domain, but also from temporal domain or fused spatio-temporal domain, which indicates that the temporal detection plays a significant role for SOD on video dataset with highly dynamic foreground objects.



(a)

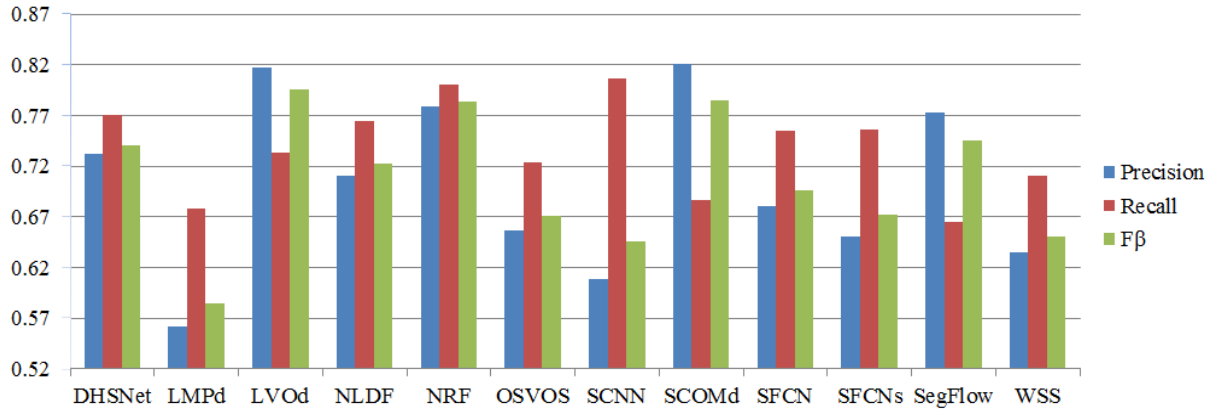


(b)

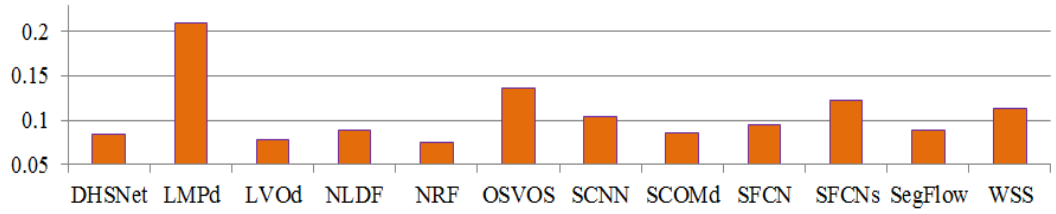


(c)

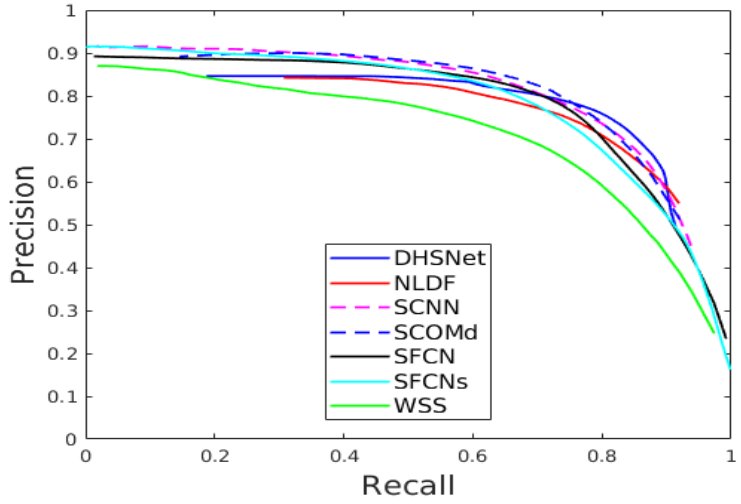
Figure 3.7: (Better viewed in color) Performances on the VOS-E dataset: (a) F-measure↑, Precision↑, Recall↑, (b) MAE↓, (c) P-R curve. ↑ means the higher the better and ↓ means the lower the better.



(a)



(b)



(c)

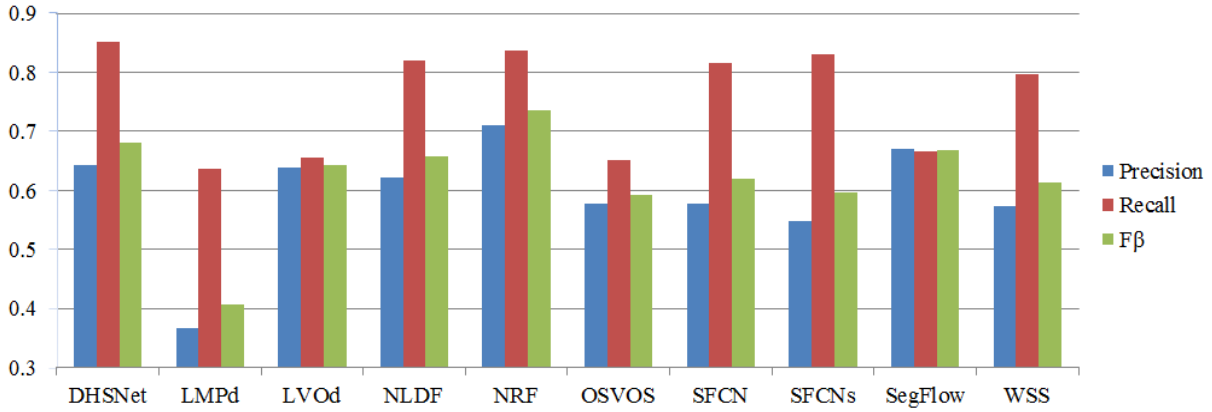
Figure 3.8: (Better viewed in color) Performances on the FBMS dataset: (a) F-measure↑, Precision↑, Recall↑, (b) MAE↓, (c) P-R curve.

Performance on the VOS-N and VOS dataset

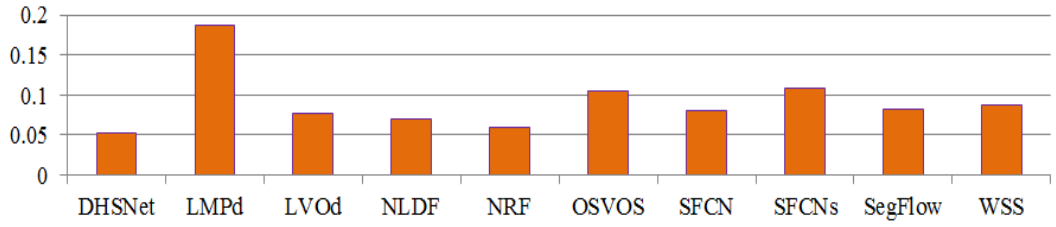
Fig. 3.9 and Fig. 3.10 show the performances on the VOS-N and the VOS datasets respectively. The VOS-N datasets contains complex scenes or highly dynamic objects, while the VOS dataset contains various cases with slow camera motion, complex scenes or highly dynamic objects. Salient objects in these two datasets are obtained according to the saliency fixation, which is similar with that in image SOD datasets. That may explain why the methods (e.g. DHSNet, NRF, NLDF, SFCN, SFCNs and WSS) trained from image SOD datasets get better results than others.

Performance on the DAVIS 2016-val and DAVIS 2017-val dataset

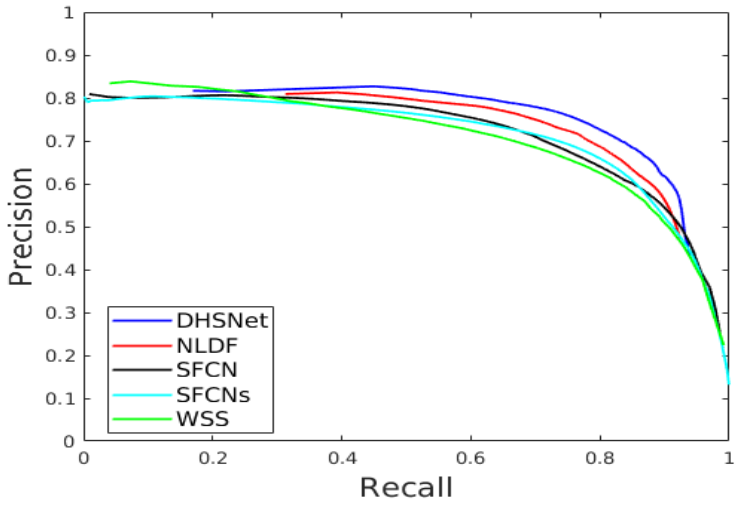
Fig. 3.11 shows the performances on the DAVIS 2016-val dataset. Fig. 3.12 shows the performances on the DAVIS 2017-val dataset. These two datasets provides multiple balanced video attributes such as occlusion, appearance change, camera-shake, etc, which help better evaluate methods' robustness. The methods that detect saliency from two domains (e.g. LVOd, NRF, SegFlow) perform better than those only from one domain (e.g. LMPd, OSVOS, WSS), which shows that saliency from two domains is more efficient for SOD on complex videos datasets. Weakly supervised methods (e.g. SCNN and WSS) get a little lower recall and F-measure values. The methods (e.g. LVOd and SegFlow) are trained from object segmentation datasets only, which shows the effectiveness of using the training datasets from closely related domains. Besides, if we compare SFCNs with SFCN, we can find that they use the same deep-learning network but with different training datasets. The input of the former one is each frame with provided ground truth, while the input of the later one is the video sequence and the detection results from SFCNs. Thus, SFCN refines the output of SFCNs, by learning more deep features from the temporal domain. If we compare LMPd and LVOd, we can find that LVOd uses the same saliency detection from temporal domain as LMPd but with extra deep spatial saliency information, and deep fused spatio-temporal features. It helps LVOd to achieve a much better performance than LMPd, which also further prove that saliency detection from two domains is significant for SOD in videos.



(a)

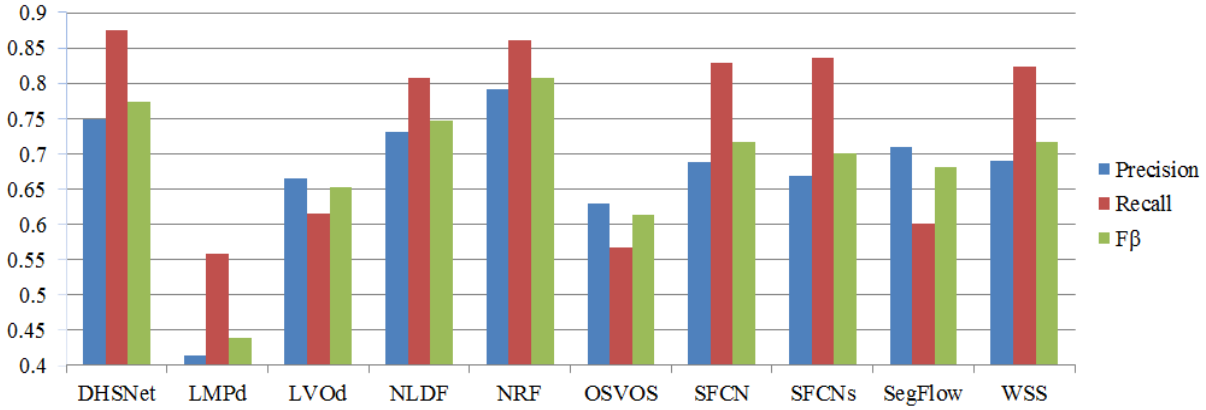


(b)

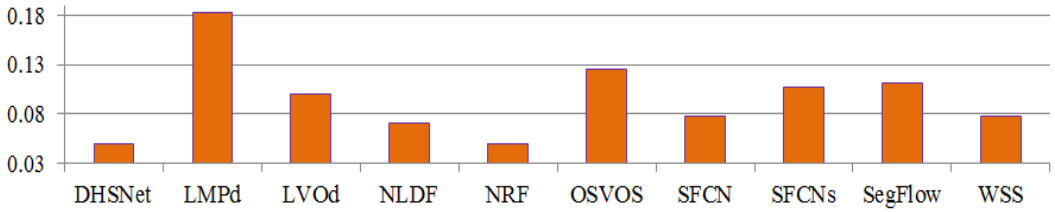


(c)

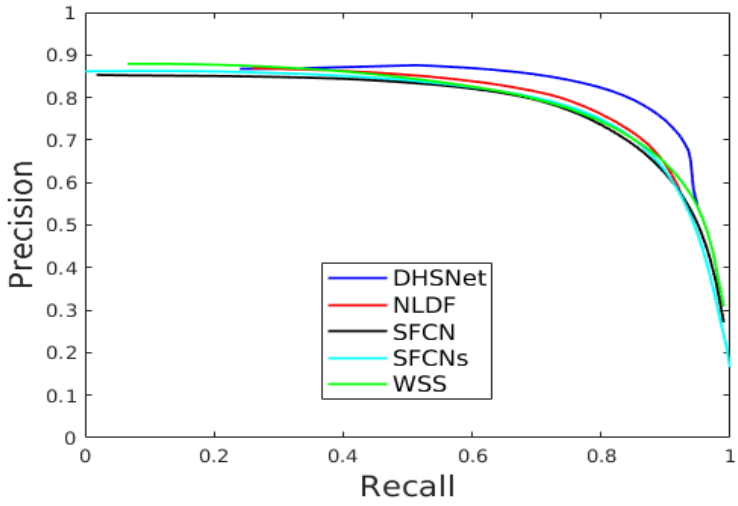
Figure 3.9: (Better viewed in color) Performances on the VOS-N dataset: (a) F-measure↑, Precision↑, Recall↑, (b) MAE↓, (c) P-R curve.



(a)

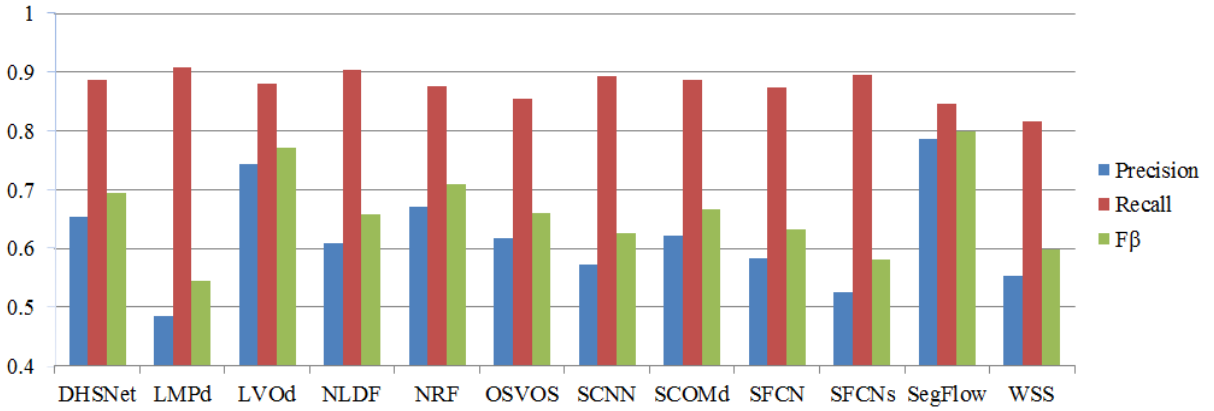


(b)

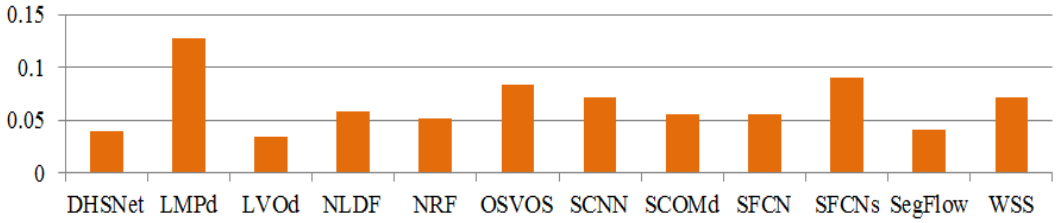


(c)

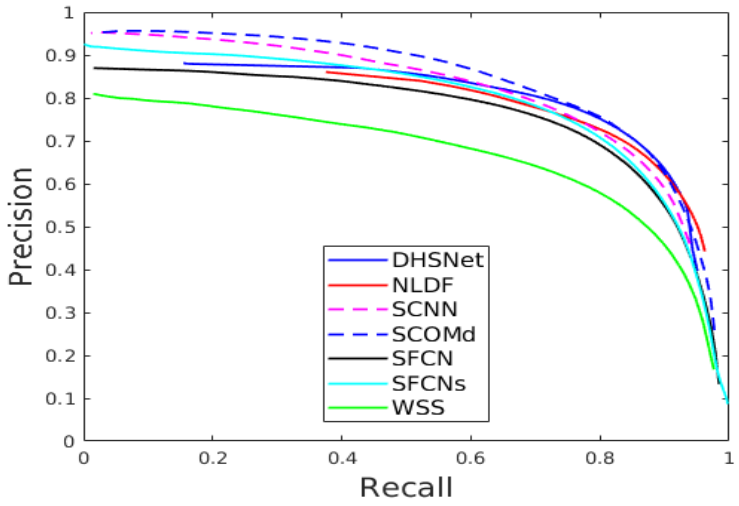
Figure 3.10: (Better viewed in color) Performances on the VOS dataset: (a) F-measure↑, Precision↑, Recall↑, (b) MAE↓, (c) P-R curve.



(a)

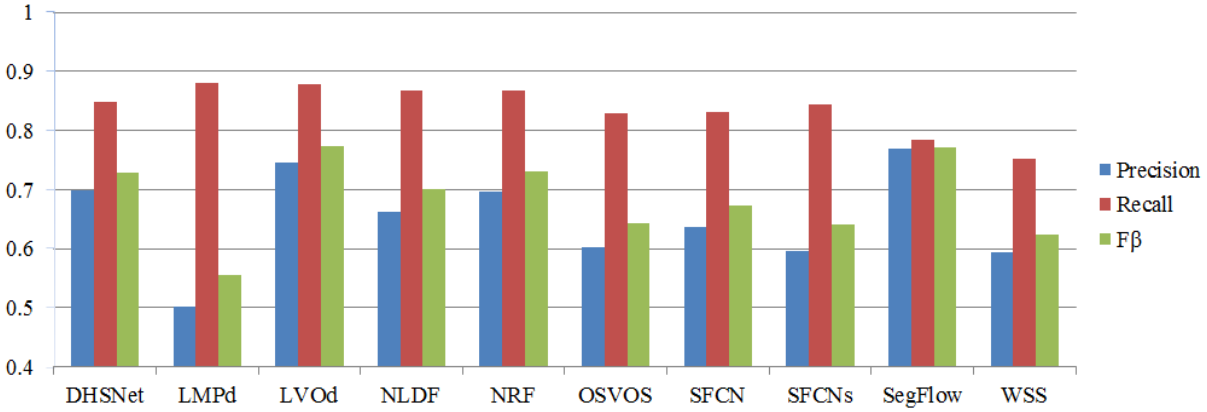


(b)

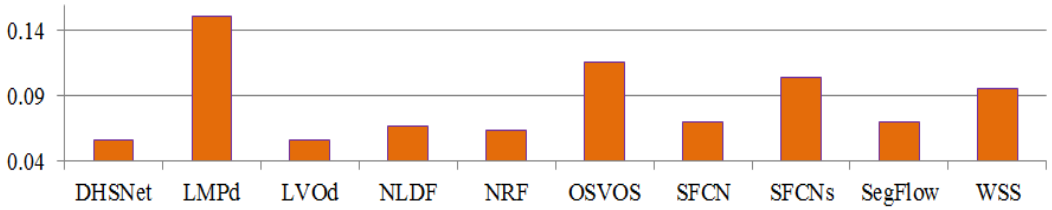


(c)

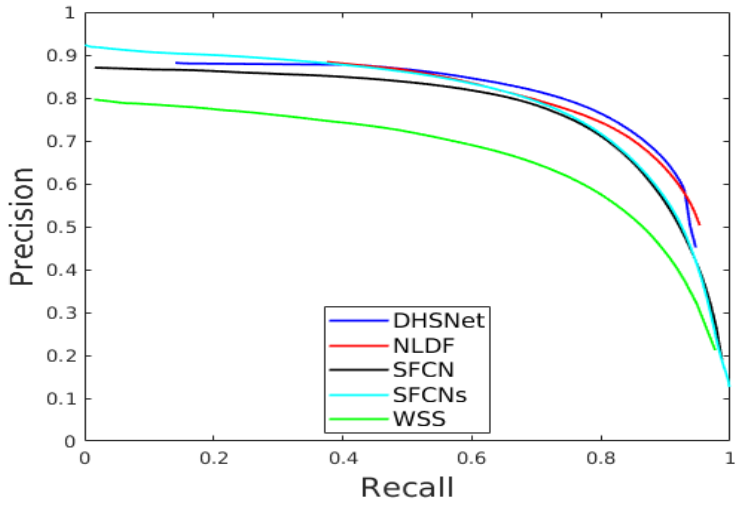
Figure 3.11: (Better viewed in color) Performances on the DAVIS-2016-val dataset: (a) F-measure↑, Precision↑, Recall↑, (b) MAE↓, (c) P-R curve.



(a)



(b)

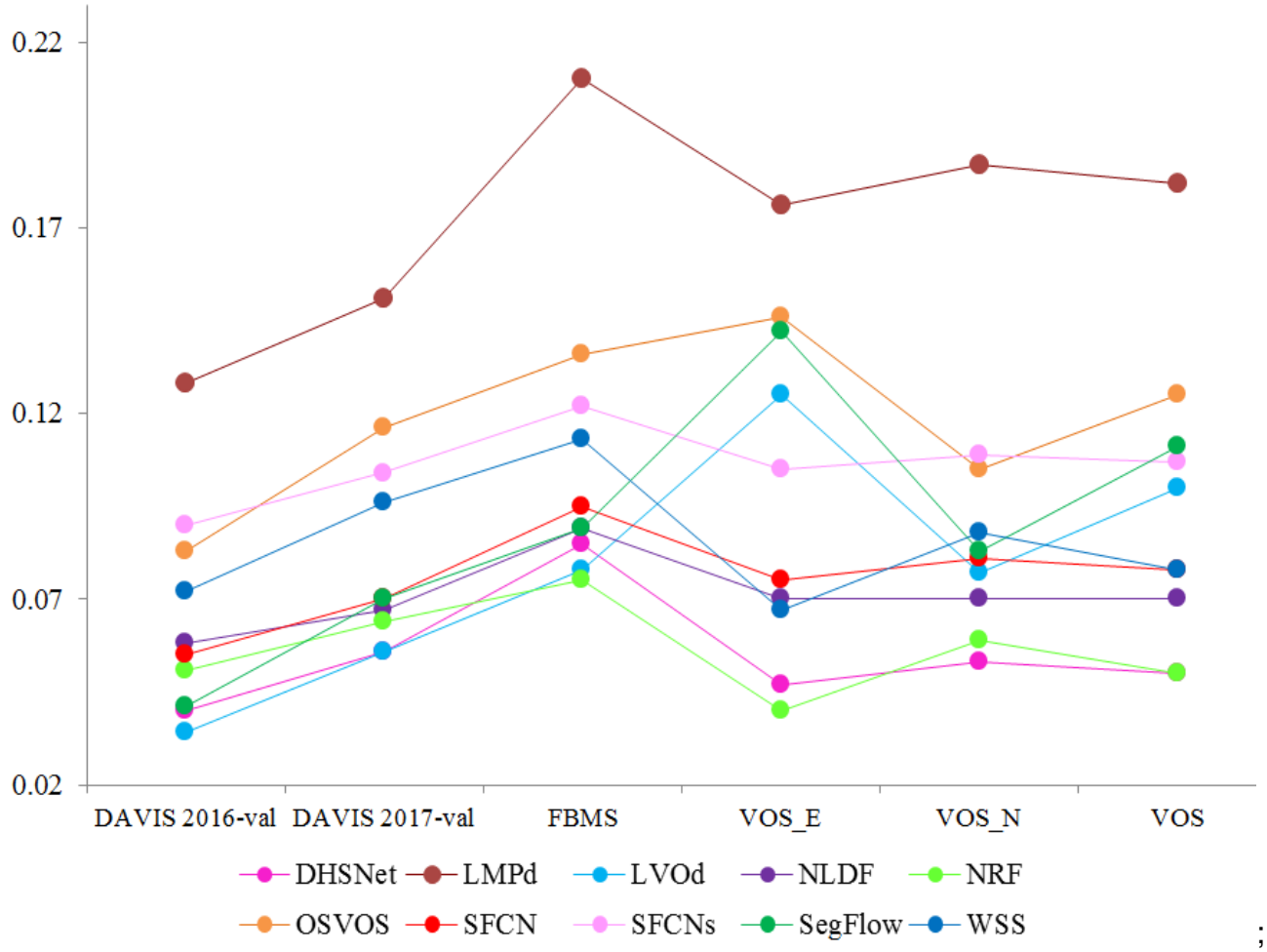


(c)

Figure 3.12: (Better viewed in color) Performances on the DAVIS-2017-val dataset: (a) F-measure↑, Precision↑, Recall↑, (b) MAE↓, (c) P-R curve

3.3.2 Global performance on various datasets

In order to catch the global view of the performance of a method on various datasets, the following Fig. 3.13 shows the comparative results of the methods for MAE metric on 6 datasets. As can be seen on this figure, methods perform less good on dataset FBMS.



;

Figure 3.13: (Better viewed in color) MAE performance↓

Fig. 3.14, Fig. 3.15 and Fig. 3.16 show the comparative results of the methods for Precision, Recall and F-measure metrics on different datasets. In each figure, the radar chart contains 10 closed curves, where each curve shows the performance of a method on the datasets. The area of the closed curve can reflect the performance of the method on the whole datasets. The larger the area the better the performance.

Table 3.3, Table 3.4 and Table 3.5 show the detailed areas of 10 curves (corresponding to the 10 methods) in Fig. 3.14, Fig. 3.15 and Fig. 3.16 respectively.

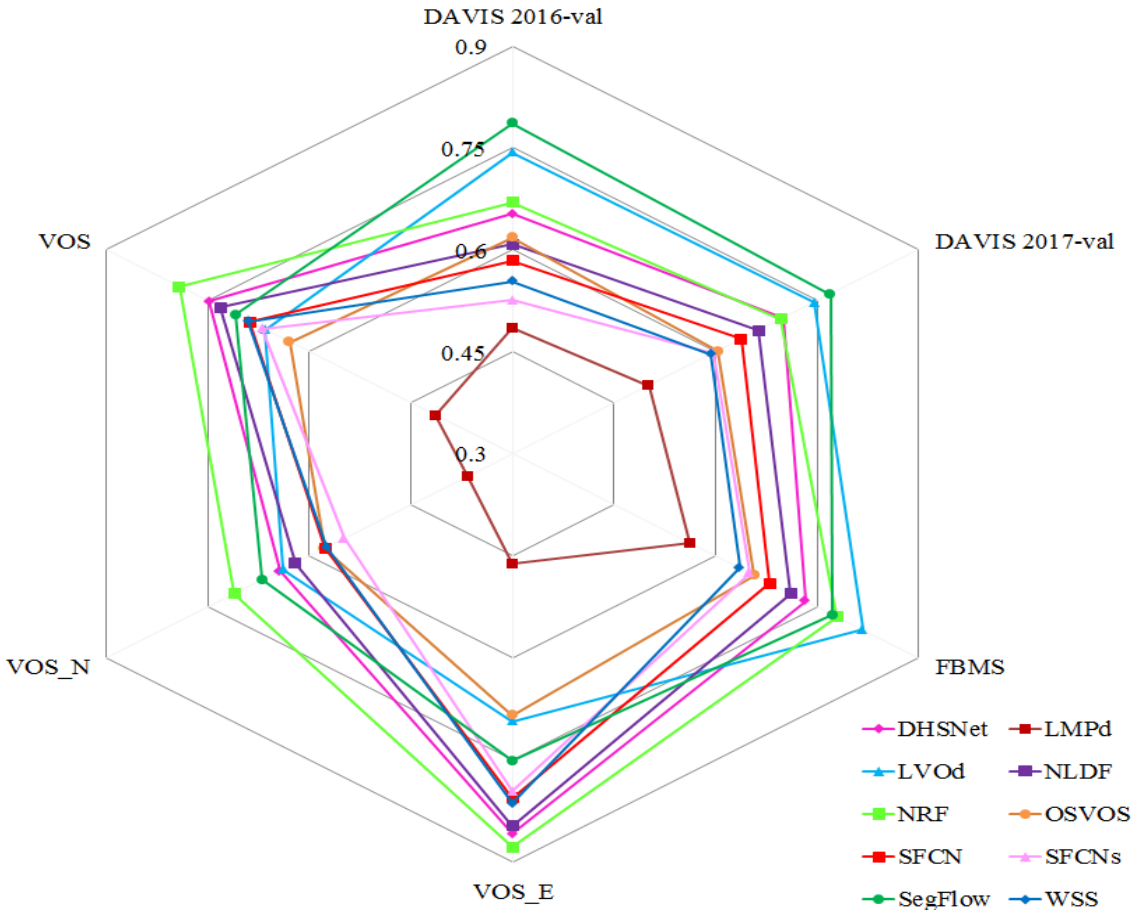


Figure 3.14: (Better viewed in color) Precision performance↑

Table 3.3: Area of each method in the Fig 3.14. (The best score is in **bold**)

DHSNet	LMPd	LVOd	NLDF	NRF	OSVOS	SFCN	SFCNs	SegFlow	WSS
1.3505	0.5650	1.3388	1.2562	**1.4768**	1.0244	1.1332	1.0282	1.4354	1.0685

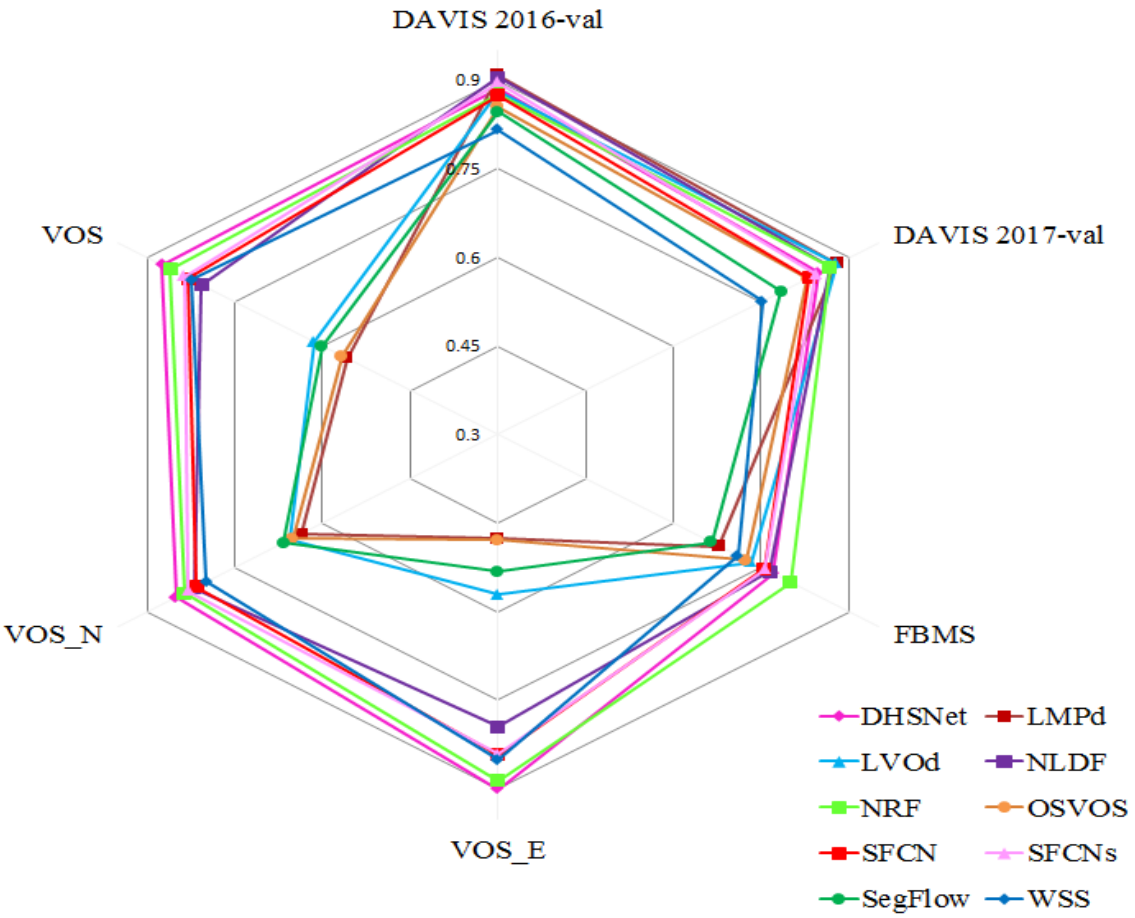


Figure 3.15: (Better viewed in color) Recall performance↑

Table 3.4: Area of each method in the Fig 3.15. (The best score is in **bold**)

DHSNet	LMPd	LVOd	NLDF	NRF	OSVOS	SFCN	SFCNs	SegFlow	WSS
1.8987	1.2482	1.3665	1.7740	1.8966	1.2212	1.7656	1.8061	1.2142	1.6278

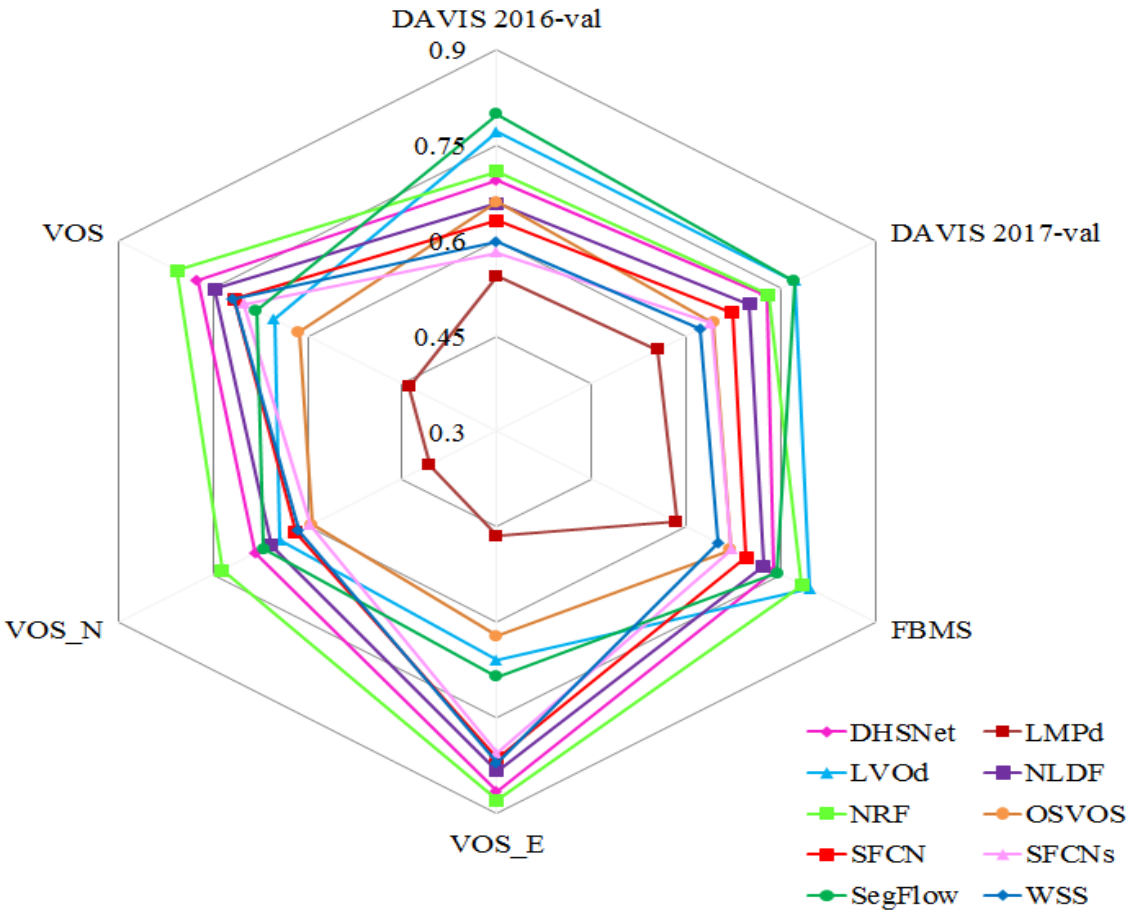


Figure 3.16: (Better viewed in color) F-measure performance↑

Table 3.5: Area of each method in the Fig 3.16. (The best score is in **bold**)

DHSNet	LMPd	LVOd	NLDF	NRF	OSVOS	SFCN	SFCNs	SegFlow	WSS
1.4477	0.6519	1.3356	1.3436	**1.5561**	1.0447	1.2384	1.1471	1.3690	1.1635

Fig. 3.14, Fig. 3.15 and Fig. 3.16 show that methods achieve highest Precision, Recall and F-measure scores on VOS-E dataset (with most static salient objects). We can learn that the deep-learning technique provides a strong ability to detect salient objects from the spatial domain. From Table 3.3, Table 3.4 and Table 3.5, one can observe that DHSNet and NRF get good Precision, Recall and F-measure scores, which are all among the best 3 scores, while LMPd performs not very well. We can firstly find that the end-to-end trainable network, DHSNet, is efficient to learn and detect the salient object; we secondly observe that though temporal saliency is significant,

saliency information only detected from the temporal domain is not enough; we thirdly note that NRF that detects the salient objects from both spatial and temporal domain is more efficient for SOD in videos.

3.3.3 Computation time comparison

A PC with a NVIDIA 1080 GPU is used for testing the speed of the methods on the DAVIS-2016-val dataset. For different models (except SCOMd and SCNN with unpublished codes), the average run-time is listed in Table 3.6.

Table 3.6: Average run time in seconds (per frame) of the compared models. (The best score is in **bold**)

Methods	DHSNet	LMPd	LVOd	NLDF	NRF	OSVOS	SFCN	SegFlow	WSS
Time(s)↓	0.069	0.2	0.42	0.091	0.297	0.072	0.072	0.174	**0.067**

From Table 3.6, we can observe that WSS has the least computation costs, which is similar to that of OSVOS, SFCN, DHSNet and NLDF. SegFlow, NRF, LMPd and LVOd are much more time-consuming.

3.3.4 Failure cases and analysis

It is difficult for all compared models to deal with some difficult cases such as the examples shown in Fig.3.17. For the first failure case, the bike is recognized with losing fine-structure by the detection network. The bikes consists of many lines, but none of them was detected and only coarse edges are shown in the final map. For the second failure case, the background object is also detected as the salient object. For the third failure case, the salient object is not detected at all.

3.4 Extension of the proposed method to integrate deep-learning technique

The above various experiments illustrate that the image-based method DHSNet gives high performance over all the tested databases. It may be interesting to look at how

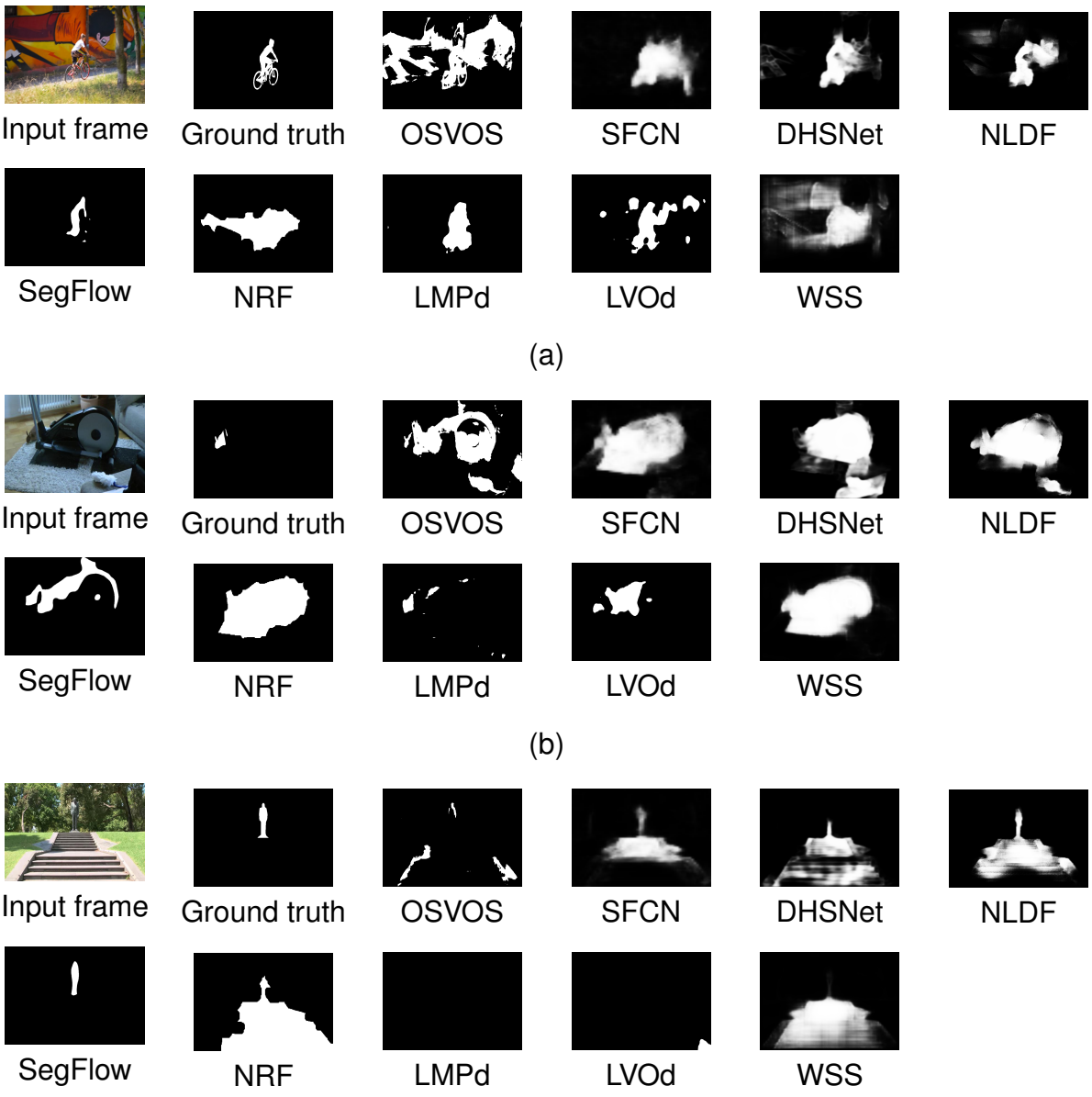


Figure 3.17: Examples of saliency maps for cases of failure.

existing traditional video SOD method can be further improved by integrating this existing deep-learning image-based SOD method. Based on this open issue, we extend our VBGF algorithm to a deep-learning method VBGFd.

3.4.1 Extension of VBGF (VBGFd)

The block-diagram of the proposed VBGFd method is shown in Fig.3.18.

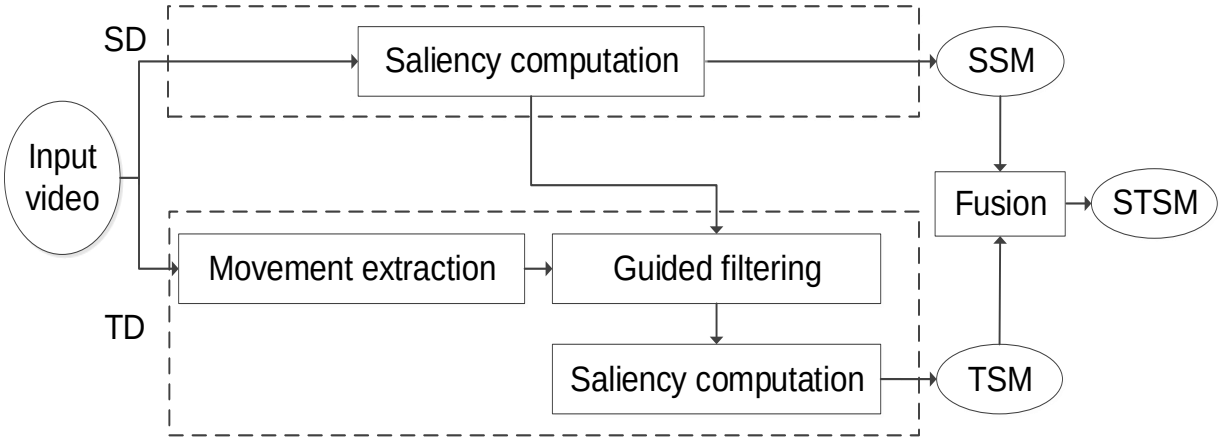


Figure 3.18: The proposed block-diagram. SD: Spatial saliency detection; SSM: Spatial saliency map; TD: Temporal saliency detection; TSM: Temporal saliency map; STSM: Spatio-temporal saliency map.

Compared with Fig.2.13, the “Virtual border building” in both “SD” and “TD” blocks is removed. The “Saliency computation” in VBGF is a traditional methods, while the “Saliency computation” in VBGFd is based on a deep-salient detection method proposed in [52]: the DHSNet (because of the availability of its source code and its good performance). In VBGFd, the first two steps in the “Map fusion” part use the ratio of the entropies for each frame in Eq.2.7.

3.4.2 Experiments and analysis

In this section, the large-scale video SOD dataset VOS and its two subsets VOS-E, VOS-N are used to show the performance of VBGFd.

Performance of components of the proposed method

The proposed VBGFd can be decomposed into different components. In Table 3.7, we list the performances of VBGFd according to its components. The 3th, 5th and

Table 3.7: Comparison of the proposed VBGFd componets' performance on dataset VOS, VOS-E, VOS-N. proSSM: proposed spatial saliency map; proTSM: proposed temporal saliency map; proSTSM: proposed spatio-temporal saliency map. The Bold number indicates the best result in each line.

Dataset	Metrics	Proposed VBGFd components			
		proSSM	proTSM without guided filtering	proTSM with guided filtering	proSTSM
VOS-E	Precision↑	0.863	0.398	0.528	**0.881**
	Recall↑	**0.905**	0.380	0.480	0.877
	F-measure↑	0.872	0.394	0.516	**0.880**
	MAE↓	0.049	0.189	0.154	**0.046**
VOS-N	Precision↑	0.649	0.407	0.407	**0.690**
	Recall↑	**0.851**	0.389	0.392	0.806
	F-measure↑	0.686	0.403	0.403	**0.714**
	MAE↓	0.055	0.136	0.132	**0.059**
VOS	Precision↑	0.753	0.403	0.466	**0.783**
	Recall↑	**0.877**	0.385	0.435	0.840
	F-measure↑	0.778	0.399	0.458	**0.795**
	MAE↓	**0.052**	0.162	0.143	0.053

6th columns show the results of the spatial saliency map, temporal saliency map and spatio-temporal saliency map. The 4th column shows the result of the temporal saliency detection without guided filtering. By comparing the 4th and 5th columns in Table 3.7, the performance is better for all performance evaluation metrics with the "guided filtering". By comparing the 3rd, 5th and 6th columns in Table 3.7, the performance is better for most evaluation metrics when the spatial saliency map and the temporal saliency map are fused together.

Performance benchmarking of the proposed method

The performance benchmarking of VBGFd, and VBGF, and 13 state-of-the-art models are reported.

In Table 3.8, Table 3.9 and Table 3.10, we inserted the performance of our proposed models into the the benchmarking table (cf. Table III in the paper [49]) provided with the VOS dataset. Note that here we only list 13 state-of-the-art models (image-based deep-learning and video-based traditional models) reported in [49]. 13 state-of-the-art models are LEGS[84], MCDL[104], MDF[48], ELD[45], DCL[47], RFCN[86],

DHSNet[52], SIV[70], FST[64], NLC[24], SAG[88], GF[89] and SSA[49]. These models are categorized into two parts: [I+D] for deep-learning and image-based, [V+U] for video-based and Unsupervised. From these three tables, we can see that among the tested 15 models, the VBGFd has the best score for 7 times, while the best benchmarked model DHSNet has the best score for 5 times. Thus in general, we can say that the VBGFd performs the best among the tested models.

Table 3.8: Performance benchmarking of VBGFd, and VBGF, and 13 state-of-the-art models on the dataset VOS-E. The best three scores in each column are marked in red, green and blue, respectively.

	Models	VOS-E			
		Precision↑	Recall↑	F-measure↑	MAE↓
[I+D]	LEGS	0.820	0.685	0.784	0.193
	MCDL	0.831	0.787	0.821	0.081
	MDF	0.740	0.848	0.762	0.100
	ELD	0.790	0.884	0.810	0.060
	DCL	0.864	0.735	0.830	0.084
	RFCN	0.834	0.820	0.831	0.075
	DHSNet	0.863	0.905	0.872	0.049
[V+U]	SIV	0.693	0.543	0.651	0.204
	FST	0.781	0.903	0.806	0.076
	NLC	0.439	0.421	0.435	0.204
	SAG	0.709	0.814	0.731	0.129
	GF	0.712	0.798	0.730	0.153
	SSA	0.875	0.776	0.850	0.062
	VBGF	0.797	0.773	0.791	0.085
	VBGFd	0.881	0.877	0.880	0.046

Computation time comparison

The deep-learning method is performed on an NVIDIA 1080 GPU, and is implemented in Python. The average run-time of the proposed VBGFd is listed in Table 3.11 in detail. Our VBGFd costs much time for exploiting optical flow (based on traditional technique), which could be accelerated by using hardware acceleration with GPU or FPGA platform, or replaced by much faster deep learning based optical flow computation method (such as FlowNet2.0). Excluding optical flow computation, our VBGFd only needs 0.34s for each frame.

Table 3.9: Performance benchmarking of VBGFd, and VBGF, and 13 state-of-the-art models on the dataset VOS-N. The best three scores in each column are marked in red, green and blue, respectively.

	Models	VOS-N			
		Precision↑	Recall↑	F-measure↑	MAE↓
[I+D]	LEGS	0.556	0.593	0.564	0.215
	MCDL	0.570	0.645	0.586	0.085
	MDF	0.527	0.742	0.565	0.098
	ELD	0.569	0.838	0.615	0.081
	DCL	0.583	0.809	0.624	0.079
	RFCN	0.614	0.783	0.646	0.080
	DHSNet	0.649	0.851	0.686	0.055
[V+U]	SIV	0.451	0.523	0.466	0.201
	FST	0.619	0.691	0.634	0.117
	NLC	0.561	0.610	0.572	0.123
	SAG	0.354	0.742	0.402	0.150
	GF	0.346	0.738	0.394	0.331
	SSA	0.660	0.682	0.665	0.103
	VBGF	0.558	0.688	0.583	0.130
	VBGFd	0.690	0.806	0.714	0.059

Table 3.10: Performance benchmarking of VBGFd, and VBGF, and 13 state-of-the-art models on the dataset VOS. The best three scores in each column are marked in red, green and blue, respectively.

	Models	VOS			
		Precision↑	Recall↑	F-measure↑	MAE↓
[I+D]	LEGS	0.684	0.638	0.673	0.204
	MCDL	0.697	0.714	0.701	0.083
	MDF	0.630	0.793	0.661	0.099
	ELD	0.676	0.861	0.712	0.071
	DCL	0.719	0.773	0.731	0.081
	RFCN	0.721	0.801	0.738	0.078
	DHSNet	0.753	0.877	0.778	0.052
[V+U]	SIV	0.568	0.533	0.560	0.203
	FST	0.697	0.794	0.718	0.097
	NLC	0.502	0.518	0.505	0.162
	SAG	0.526	0.777	0.568	0.140
	GF	0.523	0.767	0.565	0.244
	SSA	0.764	0.728	0.755	0.083
	VBGF	0.674	0.729	0.686	0.108
	VBGFd	0.783	0.840	0.795	0.053

Table 3.11: Average run time (per frame) of each component in the proposed models.

Component	VBGFd	
	Time(s)↓	Ratio(%)
saliency detection	0.15	4.78
optical flow computation	2.80	89.17
feature fusion(guided filtering)	0.07	2.23
map fusion	0.12	3.82
total	3.14	100

3.5 Conclusion

To the best of our knowledge, this is the first overview in the literature that focus on deep-learning based video SOD methods. The classification of the methods is done regarding the domain of their deep representations, which gives a new way to learn about recent development. Deep networks of some representative existing methods are introduced and compared in detail. They are surveyed from two ponts of view: frameworks and raw results. A comparative summary of methods is presented and their performances on various datasets are discussed. An effective way is presented for readers to study these 11 methods quickly. In addition, the various experiments conducted show that the methods DHSNet and NRF give high performance over all the tested databases. This overview aims to pave a way to study the existing deep-learning based video SOD methods.

We also have extended our proposed traditional video SOD method (VBGF) to VBGFd by integrating an image-based deep-learning method [52]. Various experimental results confirms combining the traditional video SOD method (VBGF) with image-based deep-learning method performs better than any individual method, and shows that compared to the tested state-of-the-art methods, VBGFd yields improved good performance.

CHAPTER 4

DEEP-LEARNING METHOD FOR SEMI-SUPERVISED VIDEO OBJECT SEGMENTATION

The video SOD in previous two chapters aims to detect salient objects from background without distinguishing each object. But, it is better for video content understanding to assign consistent object IDs to each object. This chapter focus on this task. In this chapter, Section 4.1 introduces an overview of state-of-the-art methods. Section 4.2 gives some existing issues. Section 4.3 presents the proposed method in detail. In section 4.4, we show and discuss the performances of the proposed method. Section 4.5 concludes the chapter.

4.1 An overview of state-of-the-art methods

For semi-supervised video object segmentation based on the human-guidance, one challenge is how to segment a pre-defined object in a video based on its provided mask of the frame in which the object appears at the first time. Recent works are introduced based on the way to use the human-guidance.

4.1.1 Online-offline learning

An initial way for semi-supervised video object segmentation is to firstly train the parent network which detects all foreground objects (also called as offline learning), secondly fine-tune the parent network for the particular object using the manual label (also called as online learning), as in state-of-the-art methods [9]. However, it is very time-consuming. The methods [9, 18, 66, 83] employ the combination of offline and online

learning strategies, as in Fig. 4.1 (a).

Caelles *et al.* [9] design a network to learn the foreground object, as in Fig. 4.1 (b), which is consisted of a foreground branch and a contour branch.

Compared with OSVOS [9], OnAVOS [83] updates the result based on online selected training example. It aims at adapting the changes in appearance. Fig. 4.2 presents an example.

Cheng *et al.* [18] propose a network which has two branches: the segmentation branch and the flow branch to predict the foreground objects, as in Fig. 4.3.

MaskTrack [66] predicts the segmentation mask with a rough estimated mask of the previous frame as in Fig. 4.4.

4.1.2 Mask warping

Most works adopt “mask warping” to combine the necessary appearance information and the temporal context together, which benefits the semi-supervised video object segmentation. The mask of the target object is warped to the optical flow vectors to generate warped map frame by frame [38, 51, 67, 75, 94, 95, 97].

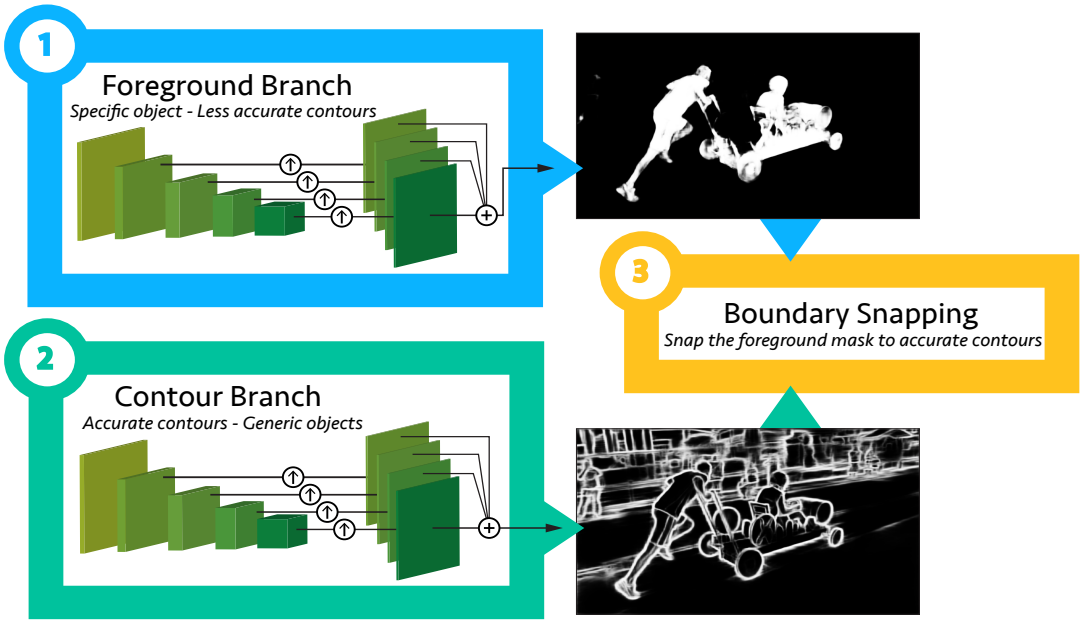
4.2 Introduction of the existing issue

“Mask warping” is faster than online learning. However, the warped map generated is vulnerable to lighting changes, deformations, etc. The wrongly detected regions in one frame can be propagated to the following ones, thus more background is warped. To solve this problem, Leibe *et al.* [55] proposed to optimize the generated warped map in each step with an objectness score etc; Khoreva *et al.* [38] proposed to optimize the generated warped map by removing the possibly spurious blobs.

The semantics label of the object instance in the first frame is another useful cue for semi-supervised video object segmentation. In the method [59], a semantics instance segmentation algorithm is leveraged to obtain the semantics label of the target object in the first frame, and then the semantics label is propagated to the following frames. In the method [40], objects are divided into human and non-human object instances which are propagated using different networks.

Mask warping and semantics label guidance are not mutually exclusive, and could be taken simultaneously. Few studies combine the advantages of the two aforemen-

(a)



(b)

Figure 4.1: OSVOS [9]. (a) An overview of OSVOS: the designed network is firstly trained to learn the generic objects, and then fine-tuned to learn the target object. (b) The designed network. (Figures are copied from the published paper [18])

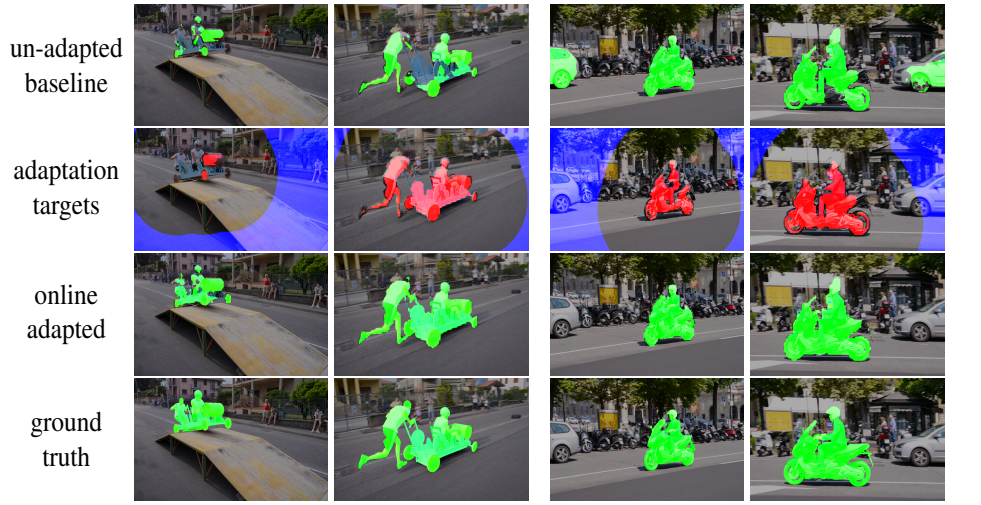


Figure 4.2: OnAVOS [83]. The first row shows the result without updating, the second row gives the online selected training example and the third shows the result with updating. (Figures are copied from the published paper [83])

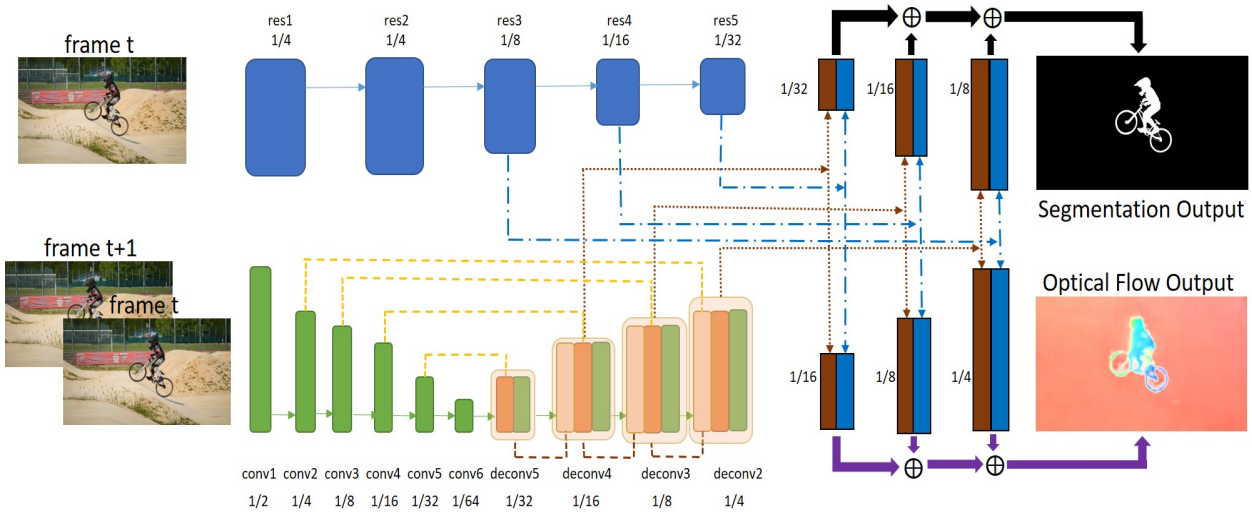


Figure 4.3: Segflow [18]. (Figures are copied from the published paper [18])

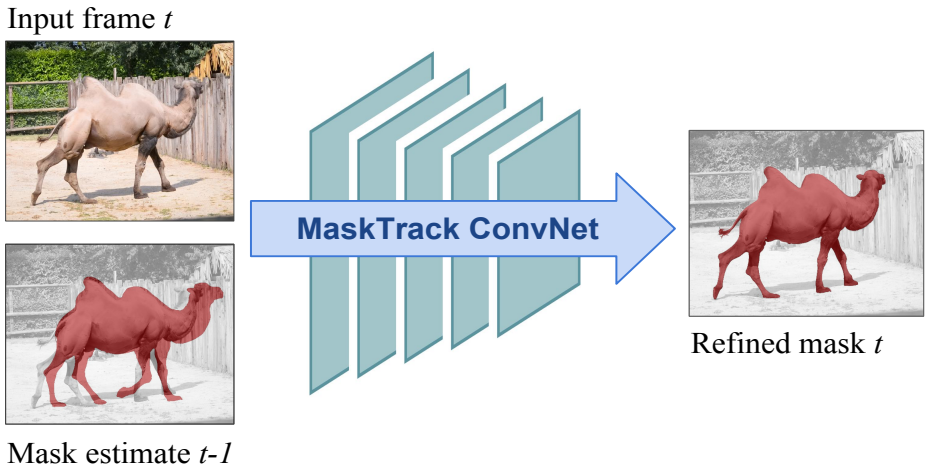


Figure 4.4: MaskTrack [66]. (Figures are copied from the published paper [66])

tioned cues. In order to take merits of mask warping and semantics label guidance, we propose approach presented hereafter a novel semi-supervised video object segmentation.

4.3 Semantic-guided warping for semi-supervised video object segmentation (SWVOS) algorithm

The proposed SWVOS consists of three main steps: (1) according to the provided pixel-wise mask of the first frame, target object is firstly segmented using mask warping technique, where warped maps are generated; (2) the warping confidence is computed for each warped map, which is then divided into high-confidence map and low-confidence map; (3) the warped map with high-confidence is directly used as the final segmentation maps, while the low-confidence warped map is optimized using semantics selection. The proposed block-diagram is shown in Fig.4.5.

4.3.1 Mask warping

The optical flow vectors between pairs of successive frames are generated using the Flownet [34]. Then the warped map of each frame is obtained by warping the proposal

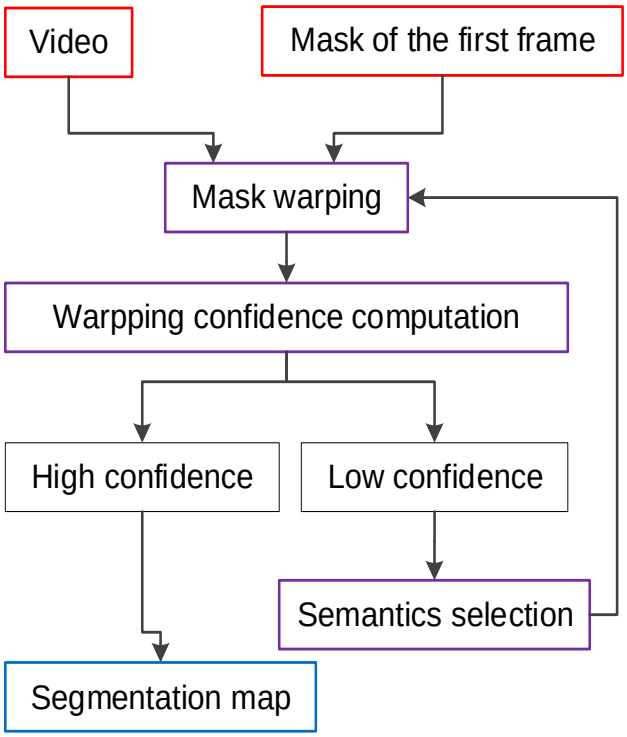


Figure 4.5: The proposed block-diagram SWVOS.

of the previous frame to the optical flow vector. The warping function is defined as:

$$f_j = \omega(f_i, V_{i \to j}) \quad j = i + 1 \tag{4.1}$$

where f_j denotes the warped map of the frame j, ω is the bilinear warping function, f_i denotes the warped map of the previous frame i (for the first frame, the proposal is the provided mask), $V_{i \to j}$ is the optical flow vectors between pairs of successive frames i and j.

4.3.2 Warping confidence computation

For the generated warped map, overlap ratio and contiguous groups number are used for warping confidence computation (WCC). Overlap ratio (OR) is the ratio of the object that belongs to the warped map (WM) and the foreground map (FM), the larger is better.

$$\mathrm{OR} = \frac{|\mathrm{WM} \bigcap \mathrm{FM}|}{|\mathrm{FM}|} \tag{4.2}$$

Contiguous groups number (CGN) is the number of contiguous regions in the warped map, the smaller is better. The warped map with a low OR value or a high CGN is regarded as low-confidence in the WCC.

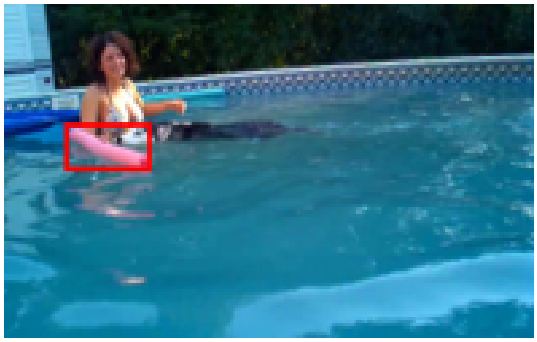
The foreground map (FM) is obtained with a fully covolutional network (FCN), which is a modified NLDF network [57]. Our FCN differs from the NLDF [57] in that (1) the NLDF resizes the input image to a fixed size while our FCN uses it with its original size; (2) the NLDF adopts the VGG [73] as the baseline and uses the output of the 5-th block in the VGG as the global feature, while our FCN removes this global feature which may bring noises for complex scenes; (3) the NLDF uses the cross entropy loss and the boundary IOU loss for training while our FCN only uses the cross entropy loss since our experiment showed that the boundary IOU loss does not influence a lot our method's performances.

One example of the WCC is given in Fig.4.6. In this example, we can see that the warped map not only contains many contiguous groups, but also has low overlap region with the foreground map. Thus, it is judged to be a warped map with low-confidence. In this chapter, the threshold for the OR is just set to be a small number 0.001. The threshold for the CGN is set to be 10, i.e. about five times of the average number of objects in each frame in the video sequence.

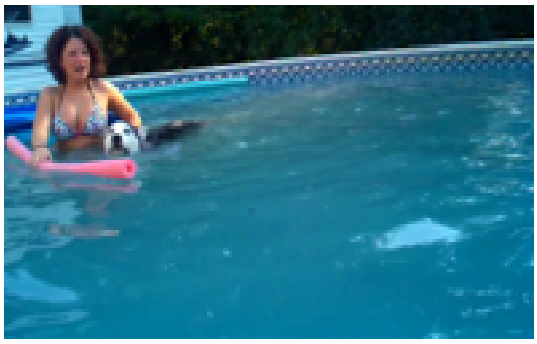
4.3.3 Semantics selection

The warped map with low-confidence is optimized using semantic selection (SS) as following. Firstly, the semantic label of the target object in the first frame is detected using the MASK R-CNN [30]. Secondly, for the frame with low-confidence warped map, semantics of all objects are detected using the MASK R-CNN. Thirdly, the object in the frame that satisfies two conditions is segmented to generate the optimized warped map: (1) the object has the same semantic label as the target object, (2) the object is the closest one to the center of gravity of the low-confidence warped map. Here the MASK R-CNN is fine-tuned with the YouTube-VOS-train dataset [96] in order to recognize categories in this dataset which has much more classes than the previous datasets. One example is given in Fig.4.7.

For a video sequence with multiple pre-defined objects, these target objects are



(a) Pre-defined target object in the 1st frame

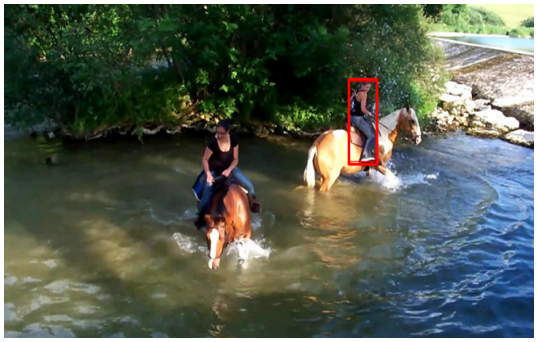


(b) Input frame

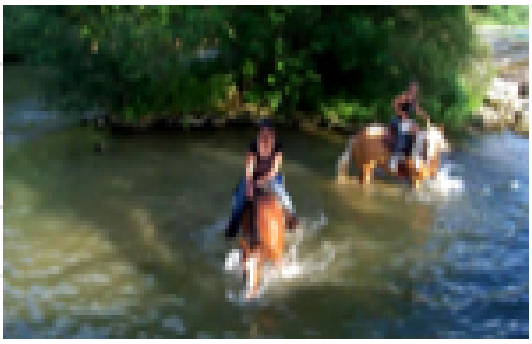
(c) Warped map

(d) Foreground map

Figure 4.6: One example of the warping confidence computation. The target object is denoted in red box in (a).

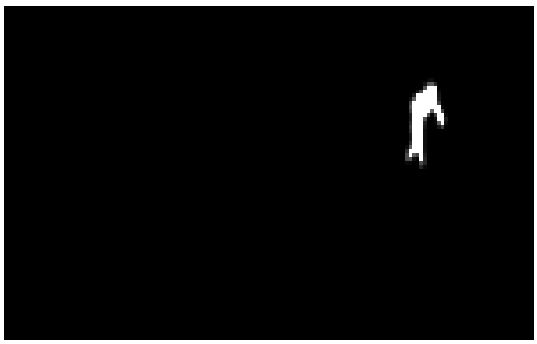


(a) Pre-defined target object in the 1st frame



(b) Input frame

(c) Warped map before SS



(d) Warped map after SS

Figure 4.7: One example of semantics selection (SS). The target object is denoted in red box in (a).

detected separately, and then merged together to generate the final segmentation map. If the pixel is detected belonging to multiple target objects, it is set to the one that has the smallest size in the provided manual labels in the first frame.

4.4 Experiments and analyses

This section shows the performances of our approach. Table.4.1 compares our pro-

Table 4.1: Performance comparison between the proposed method (SWVOS) and existing models over the YouTube-VOS-test dataset. The best score is in **bold**.

Methods	J_seen↑	J_unseen↑	F_seen↑	F_unseen↑	Overall↑
OnAVOS	0.557	0.568	0.613	0.623	0.590
MaskTrack	0.569	**0.607**	0.593	0.637	0.602
OSVOS	**0.591**	0.588	**0.637**	**0.639**	**0.614**
SWVOS	0.513	0.367	0.494	0.419	0.448
Segflow	0.404	0.385	0.350	0.327	0.367

posed method with the state-of-the-art methods. We can see that the proposed method achieves the better performance than Segflow [18] on the YouTube-VOS-test dataset. We must note that the compared methods OSVOS, OnAVOS and MaskTrack perform better than our proposed method. However they all use the time-consuming online learning step, which is not suitable for real-world applications. Our proposed method has not this limitation.

For the semi-supervised video object segmentation task, the YouTube-VOS Challenge on video object segmentation 2018 use YouTube-VOS-test dataset for competition. Our method achieves the 8th result in YouTube-VOS Challenge on video object segmentation 2018. In Table 4.2, we show the performance of our proposed models (named “SnowFlower”) in the benchmarking table. Note that only 8 models are selected and listed.

4.5 Conclusion

In this chapter, we have proposed a novel semi-supervised video object segmentation method that extracts each pre-defined object from each frame. This goal is achieved

Table 4.2: Performance benchmarking in the YouTube-VOS Challenge.

Team Name	Overall	J_seen	J_unseen	F_seen	F_unseen	Rank
Jono	0.722(1)	0.737(1)	0.648(2)	0.778(1)	0.725(2)	1st
speeding_zZ	0.720(2)	0.725(3)	0.663(1)	0.752(3)	0.741(1)	2nd
mikirui	0.699(3)	0.736(2)	0.621(4)	0.755(2)	0.684(4)	3rd
hi.nine	0.684(4)	0.706(5)	0.623(3)	0.728(5)	0.677(5)	4th
sunpeng	0.672(5)	0.707(4)	0.598(6)	0.736(4)	0.648(6)	5th
random_name	0.672(6)	0.672(6)	0.609(5)	0.709(6)	0.697(3)	6th
kduarte	0.539(7)	0.594(7)	0.483(7)	0.578(7)	0.502(7)	7th
SnowFlower	0.448(8)	0.513(8)	0.367(8)	0.494(8)	0.419(8)	8th

by using the mask warping technique. By employing the warping confidence computation, the method can firstly detect the warped mask in low-level confidence. Then the optimized warped flow map is achieved through re-identifying the target object with semantics selection. The wrong segmented regions in the warped map is alleviated and the target object is extracted with better performance. For the evaluation of video object segmentation, a recently published large-scale dataset: Youtube-VOS is used. Experimental results demonstrate that the proposed method achieves high J value and F value.

CHAPTER 5

CONCLUSION AND PERSPECTIVE

This thesis focuses on the problems of video salient object detection (SOD) that aim at separating salient object from background in each frame of a video sequence and the problems of semi-supervised video object segmentation that aim at assigning consistent object IDs to each pixel in each frame of a video sequence. We have proposed a traditional method for video SOD, an overview of deep-learning methods for video SOD, an extension of the proposed traditional method to integrate deep-learning and a deep-learning method for semi-supervised video object segmentation as follows:

-The proposed traditional method for video SOD (VBGF) is based on “background prior”, which takes the frame boundary as the background. The virtual border building is proposed to detect the salient object that touches the frame border. A “Feature fusion” is employed to enhance the detected salient object edges from the temporal domain. A “Map fusion” is used to combine the SSM and TSM together to generate the final saliency result. We have compared our video SOD model with the state-of-the-art models and the experiments demonstrate that the proposed model obtains significant improvement over the state-of-the-art approaches.

-The survey of the video SOD puts emphasis on deep-learning based methods in this domain. This survey firstly aims at classifying the existing methods and analyze their frameworks, which may benefit the future work. This survey secondly aims at making a comparison of the performances of the state-of-the-art methods. We used four popular datasets and five commonly used evaluation metrics. The results shows that the methods DHSNet and NRF performance good over all the tested databases.

-The extended model VBGFd is motivated by the observation that deep-learning image SOD achieves a good performance to detect the salient object from spatial domain. Combining deep representations from image SOD task helps to detect the salient object in videos. We have carried out evaluations on a large benchmarking dataset and experiments demonstrated the extended model achieves the state-of-the-art performance.

-The proposed video object segmentation model SWVOS, based on deep learning techniques, uses the semantics of the object as a guidance during the warping process. Experimental results on a large-scale dataset Youtube-VOS demonstrate that the proposed method achieves good performance.

Some future works can be derived from the previous analyses:

- Employ some more useful deep representations: the guided filter used in VBGF may lead to information loss as the used hand-crafted features are not robust in some complex cases, which may be improved with informative deep representation features.

- Train some deep network for the map fusion: although the map fusion in VBGF based on traditional methods gives a good balance between the SSM and the TSM, it makes some failures when salient objects have not distinct appearance and motion information at the same time. It would be interesting to verify that the Map fusion method in VBGF can improve by using different deep networks.

- Employ more video saliency cues: it is valuable to investigate for other deep representations that can improve the quality of video saliency detection. The image object-level cue used in the VBGFd is the most popular choice. Human visual attention usually pays more attention on certain categories, thus the object classification cue can be considered as another choice to detect the video SOD.

- Explore more temporal saliency features and spatio-temporal saliency features: from our experiments, deep-learning technique performs well for detecting the salient object from the spatial domain. Most of the existing video SOD mainly rely on the spatial saliency detection and based on a backbone network. However the goal of video SOD is to detect the object which is salient in the whole video sequence. Further exploration for the temporal saliency features and spatio-temporal saliency features need to be explored.

- Explore weakly-supervised networks [76]: fully supervised models improve detection performance but rely on large training dataset with provided ground truth. Weakly-supervised models that do not rely on large pixel-wise labels attract much attention in recent years. However, its accuracy is still far from satisfactory, and further accuracy improving is one topic to investigate in the future.

-In video salient object detection, the video salient object may change (also called as saliency shift), which is challenging and firstly pointed out in recent method [25]. Due to the dynamic human attention characteristics, considering such saliency shift is more realistic and is helpful for video understanding.

-In semi-supervised video object segmentation, the mask of the object is given. However, the mask needs a pixel-level accurate segmentation, which is time-consuming. Interactive segmentation using scribble supervision [10] is proposed recently. The user is asked to draw scribbles on the object instance, in order to refine the output of a method interactively until the result is satisfactory. In order to further decrease the human supervision, the unsupervised objects instance segmentation [11], which does not take any user input into account, are more attractive.

LIST OF ABBREVIATIONS

ASPP	“à trous” pyramid pooling.
BM	binary mask.
CRF	conditional random field.
DDB	divided down border.
DLB	divided left border.
DRB	divided right border.
DUB	divided up border.
F	Contour Accuracy.
FastMBD	Fast iterative Minimum barrier distance transform algorithm.
FBMS	Freiburg-Berkeley Motion Segmentation.
FCN	fully convolutional networks.
GT	Ground truth.
IOU	Intersection over Union.
J	Region Similarity.
MAE	Mean Absolute Error.
P-R	Precision-Recall.
proSSM	proposed spatial saliency map.
proSTSM	proposed spatial-temporal saliency map.
proTSM	proposed temporal saliency map.

SAD	sum of absolute differences.
SD	Spatial saliency detection.
SM	saliency map.
SOD	Salient object detection.
SSM	Spatial saliency map.
STSM	Spatio-temporal saliency map.
SWVOS	Semantic-guided warping for semi-supervised video object segmentation.
TD	Temporal saliency detection.
TSM	Temporal saliency map.
VBGF	Virtual Border and Guided Filter-based salient object detection for videos.
VBGFd	extension of the VBGF.
VDB	the virtual down border.
VLB	virtual left border.
VRB	virtual right border.
VUB	virtual up border.

LIST OF PUBLICATION

- “Fast filtering-based temporal saliency detection using Minimum Barrier Distance”, Qiong Wang, Lu Zhang, and Kidiyo Kpalma. In: 2017 IEEE International Conference on Multimedia & Expo Workshops, ICME Workshops, Hong Kong, China, July 10-14, 2017. 2017, pp. 232–237.
- “An Semantic-guided warping for semi-supervised video object segmentation”, Qiong WANG, Lu Zhang and Kidiyo Kpalma. The 1st Large-scale Video Object Segmentation Challenge Workshop in conjunction with ECCV2018, Munich, Germany.

BIBLIOGRAPHY

[1] Çaglar Aytekin, Horst Possegger, Thomas Mauthner, Serkan Kiranyaz, Horst Bischof, and Moncef Gabbouj. “Spatiotemporal Saliency Estimation by Spectral Foreground Detection”. In: *IEEE Trans. Multimedia* 20.1 (2018), pp. 82–95.

[2] Simon Baker, Daniel Scharstein, J. P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski. “A Database and Evaluation Methodology for Optical Flow”. In: *International Journal of Computer Vision* 92.1 (2011), pp. 1–31.

[3] Saumik Bhattacharya, Venkatesh K. Subramanian, and Sumana Gupta. “Visual Saliency Detection Using Spatiotemporal Decomposition”. In: *IEEE Trans. Image Processing* 27.4 (2018), pp. 1665–1675.

[4] Xiu-Li Bi and Chi-Man Pun. “Fast copy-move forgery detection using local bidirectional coherency error refinement”. In: *Pattern Recognition* 81 (2018), pp. 161–175.

[5] Ali Borji, Ming-Ming Cheng, Huaizu Jiang, and Jia Li. “Salient Object Detection: A Benchmark”. In: *IEEE Trans. Image Processing* 24.12 (2015), pp. 5706–5722.

[6] Ali Borji, Ming-Ming Cheng, Huaizu Jiang, and Jia Li. “Salient Object Detection: A Survey”. In: *arXiv* abs/1411.5878 (2014).

[7] Thomas Brox and Jitendra Malik. “Large Displacement Optical Flow: Descriptor Matching in Variational Motion Estimation”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 33.3 (2011), pp. 500–513.

[8] Thomas Brox and Jitendra Malik. “Object Segmentation by Long Term Analysis of Point Trajectories”. In: *Computer Vision - ECCV 2010 - 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part V*. 2010, pp. 282–295.

[9] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. “One-Shot Video Object Segmentation”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. 2017, pp. 5320–5329.

[10] Sergi Caelles, Alberto Montes, Kevis-Kokitsi Maninis, Yuhua Chen, Luc Van Gool, Federico Perazzi, and Jordi Pont-Tuset. “The 2018 DAVIS Challenge on Video Object Segmentation”. In: *CoRR* abs/1803.00557 (2018).

[11] Sergi Caelles, Jordi Pont-Tuset, Federico Perazzi, Alberto Montes, Kevis-Kokitsi Maninis, and Luc Van Gool. “The 2019 DAVIS Challenge on VOS: Unsupervised Multi-Object Segmentation”. In: *CoRR* abs/1905.00737 (2019).

[12] Ioannis Cassagne, Nicolas Riche, Marc Decombas, Matei Mancas, Bernard Gosselin, Thierry Dutoit, and Robert Laganière. “Video saliency based on rarity prediction: Hyperaptor”. In: *23rd European Signal Processing Conference, EUSIPCO 2015, Nice, France, August 31 - September 4, 2015*. 2015, pp. 1521–1525.

[13] Chenglizhao Chen, Shuai Li, Hong Qin, Zhenkuan Pan, and Guowei Yang. “Bilevel Feature Learning for Video Saliency Detection”. In: *IEEE Trans. Multimedia* 20.12 (2018), pp. 3324–3336.

[14] Chenglizhao Chen, Shuai Li, Yongguang Wang, Hong Qin, and Aimin Hao. “Video Saliency Detection via Spatial-Temporal Fusion and Low-Rank Coherency Diffusion”. In: *IEEE Trans. Image Processing* 26.7 (2017), pp. 3156–3170.

[15] Chenglizhao Chen, Yunxiao Li, Shuai Li, Hong Qin, and Aimin Hao. “A Novel Bottom-Up Saliency Detection Method for Video With Dynamic Background”. In: *IEEE Signal Process. Lett.* 25.2 (2018), pp. 154–158.

[16] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 40.4 (2018), pp. 834–848.

[17] Yuhuan Chen, Wenbin Zou, Yi Tang, Xia Li, Chen Xu, and Nikos Komodakis. “SCOM: Spatiotemporal Constrained Optimization for Salient Object Detection”. In: *IEEE Trans. Image Processing* 27.7 (2018), pp. 3345–3357.

[18] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. “SegFlow: Joint Learning for Video Object Segmentation and Optical Flow”. In: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. 2017, pp. 686–695.

[19] Runmin Cong, Jianjun Lei, Huazhu Fu, Ming-Ming Cheng, Weisi Lin, and Qingming Huang. “Review of Visual Saliency Detection with Comprehensive Information”. In: *CoRR* abs/1803.03391 (2018).

[20] Marc Decombas, Nicolas Riche, Frédéric Dufaux, Béatrice Pesquet-Popescu, Matei Mancas, Bernard Gosselin, and Thierry Dutoit. “Spatio-temporal saliency based on rare model”. In: *IEEE International Conference on Image Processing, ICIP 2013, Melbourne, Australia, September 15-18, 2013*. 2013, pp. 3451–3455.

[21] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Häusser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. “FlowNet: Learning Optical Flow with Convolutional Networks”. In: *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. 2015, pp. 2758–2766.

[22] Lijuan Duan, Tao Xi, Song Cui, Honggang Qi, and Alan C. Bovik. “A spatiotemporal weighted dissimilarity-based method for video saliency detection”. In: *Sig. Proc.: Image Comm.* 38 (2015), pp. 45–56.

[23] Salehe Erfanian Ebadi and Ebroul Izquierdo. “Foreground Segmentation with Tree-Structured Sparse RPCA”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 40.9 (2018), pp. 2273–2280.

[24] Alon Faktor and Michal Irani. “Video Segmentation by Non-Local Consensus voting”. In: *British Machine Vision Conference, BMVC 2014, Nottingham, UK, September 1-5, 2014*. 2014.

[25] DengPing Fan, Wenguan Wang, MingMing Cheng, and Jianbing Shen. “Shifting more attention to video salient object detection”. In: *2019 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*. 2019, pp. 8554–8564.

[26] Yuming Fang, Zhou Wang, Weisi Lin, and Zhijun Fang. “Video Saliency Incorporating Spatiotemporal Cues and Uncertainty Weighting”. In: *IEEE Trans. Image Processing* 23.9 (2014), pp. 3910–3921.

[27] Ken Fukuchi, Kouji Miyazato, Akisato Kimura, Shigeru Takagi, and Junji Yamato. “Saliency-based video segmentation with graph cuts and sequentially updated priors”. In: *Proceedings of the 2009 IEEE International Conference on Multime-*

dia and Expo, ICME 2009, June 28 - July 2, 2009, New York City, NY, USA. 2009, pp. 638–641.

[28] Fang Guo, Wenguan Wang, Jianbing Shen, Ling Shao, Jian Yang, Dacheng Tao, and Yuan Yan Tang. “Video Saliency Detection Using Object Proposals”. In: *IEEE Trans. Cybernetics* 48.11 (2018), pp. 3159–3170.

[29] Junwei Han, Dingwen Zhang, Gong Cheng, Nian Liu, and Dong Xu. “Advanced Deep-Learning Techniques for Salient and Category-Specific Object Detection: A Survey”. In: *IEEE Signal Process. Mag.* 35.1 (2018), pp. 84–100.

[30] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. “Mask R-CNN”. In: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. 2017, pp. 2980–2988.

[31] Kaiming He, Jian Sun, and Xiaoou Tang. “Guided Image Filtering”. In: *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I*. 2010, pp. 1–14.

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. 2016, pp. 770–778.

[33] Yinlin Hu, Rui Song, and Yunsong Li. “Efficient Coarse-to-Fine Patch Match for Large Displacement Optical Flow”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. 2016, pp. 5704–5712.

[34] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. “FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. 2017, pp. 1647–1655.

[35] Sergey Ioffe and Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*. 2015, pp. 448–456.

[36] Bowen Jiang, Lihe Zhang, Huchuan Lu, Chuan Yang, and Ming-Hsuan Yang. “Saliency Detection via Absorbing Markov Chain”. In: *2013 IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*. 2013, pp. 1665–1672.

[37] Rajkumar Kannan, Gheorghita Ghinea, and Sridhar Swaminathan. “Discovering salient objects from videos using spatiotemporal salient region detection”. In: *Sig. Proc.: Image Comm.* 36 (2015), pp. 154–178.

[38] Anna Khoreva, Benenson Benenson, Eddy Ilg, Thomas Brox, and Bernt Schiele. “Lucid Data Dreaming for Video Object Segmentation”. In: *The 2018 DAVIS Challenge on Video Object Segmentation - CVPR Workshops* (2018).

[39] Hansang Kim, Youngbae Kim, Jae-Young Sim, and Chang-Su Kim. “Spatiotemporal Saliency Detection for Video Sequences Based on Random Walk With Restart”. In: *IEEE Trans. Image Processing* 24.8 (2015), pp. 2552–2564.

[40] T.-N. Le, K.-T. Nguyen, M.-H. Nguyen-Phan, T.-V. Ton, T.-A. Nguyen (2), X.-S. Trinh, Q.-H. Dinh, V.-T. Nguyen, A.-D. Duong, A. Sugimoto, T. V. Nguyen, and M.-T. Tran. “Instance Re-Identification Flow for Video Object Segmentation”. In: 2017.

[41] Trung-Nghia Le and Akihiro Sugimoto. “Deeply Supervised 3D Recurrent FCN for Salient Object Detection in Videos”. In: *British Machine Vision Conference 2017, BMVC 2017, London, UK, September 4-7, 2017*. 2017.

[42] Trung-Nghia Le and Akihiro Sugimoto. “Semantic Instance Meets Salient Object: Study on Video Semantic Salient Instance Segmentation”. In: *IEEE Winter Conference on Applications of Computer Vision*. 2019.

[43] Trung-Nghia Le and Akihiro Sugimoto. “SpatioTemporal utilization of deep features for video saliency detection”. In: *2017 IEEE International Conference on Multimedia & Expo Workshops, ICME Workshops, Hong Kong, China, July 10-14, 2017*. 2017, pp. 465–470.

[44] Trung-Nghia Le and Akihiro Sugimoto. “Video Salient Object Detection Using Spatiotemporal Deep Features”. In: *IEEE Trans. Image Processing* 27.10 (2018), pp. 5002–5015.

[45] Gayoung Lee, Yu-Wing Tai, and Junmo Kim. “Deep Saliency with Encoded Low Level Distance Map and High Level Features”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. 2016, pp. 660–668.

[46] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M. Rehg. “Video Segmentation by Tracking Many Figure-Ground Segments”. In: *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*. 2013, pp. 2192–2199.

[47] Guanbin Li and Yizhou Yu. “Deep Contrast Learning for Salient Object Detection”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. 2016, pp. 478–487.

[48] Guanbin Li and Yizhou Yu. “Visual Saliency Detection Based on Multiscale Deep CNN Features”. In: *IEEE Trans. Image Processing* 25.11 (2016), pp. 5012–5024.

[49] Jia Li, Changqun Xia, and Xiaowu Chen. “A Benchmark Dataset and Saliency-Guided Stacked Autoencoders for Video-Based Salient Object Detection”. In: *IEEE Trans. Image Processing* 27.1 (2018), pp. 349–364.

[50] Jia Li, Anlin Zheng, Xiaowu Chen, and Bin Zhou. “Primary Video Object Segmentation via Complementary CNNs and Neighborhood Reversible Flow”. In: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. 2017, pp. 1426–1434.

[51] Xiaoxiao Li, Yuankai Qi, Zhe Wang, Kai Chen, Ziwei Liu, Jianping Shi, Ping Luo, Change Loy Chen, and Xiaoou Tang. “Video Object Segmentation with Re-identification”. In: *The 2017 DAVIS Challenge on Video Object Segmentation - CVPR Workshops* (2017).

[52] Nian Liu and Junwei Han. “DHSNet: Deep Hierarchical Saliency Network for Salient Object Detection”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. 2016, pp. 678–686.

[53] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum. “Learning to Detect a Salient Object”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 33.2 (2011), pp. 353–367.

[54] Zhi Liu, Junhao Li, Linwei Ye, Guangling Sun, and Liquan Shen. “Saliency Detection for Unconstrained Videos Using Superpixel-Level Graph and Spatiotemporal Propagation”. In: *IEEE Trans. Circuits Syst. Video Techn.* 27.12 (2017), pp. 2527–2542.

[55] Jonathon Luiten, Paul Voigtlaender, and Bastian Leibe. “PReMVOS: Proposal-generation, Refinement and Merging for the DAVIS Challenge on Video Object Segmentation 2018”. In: *The 2018 DAVIS Challenge on Video Object Segmentation - CVPR Workshops* (2018).

[56] Ye Luo, Junsong Yuan, and Jianwei Lu. “Finding spatio-temporal salient paths for video objects discovery”. In: *J. Visual Communication and Image Representation* 38 (2016), pp. 45–54.

[57] Zhiming Luo, Akshaya Kumar Mishra, Andrew Achkar, Justin A. Eichel, Shaozi Li, and Pierre-Marc Jodoin. “Non-local Deep Features for Salient Object Detection”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. 2017, pp. 6593–6601.

[58] Muhammad Habib Mahmood, Yago Diez, Joaquim Salvi, and Xavier Lladó. “A collection of challenging motion segmentation benchmark datasets”. In: *Pattern Recognition* 61 (2017), pp. 1–14.

[59] Kevis-Kokitsi Maninis, Sergi Caelles, Yuhua Chen, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. “Video Object Segmentation Without Temporal Information”. In: *arXiv* abs/1709.06031 (2017).

[60] Nikolaus Mayer, Eddy Ilg, Philip Häusser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. “A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. 2016, pp. 4040–4048.

[61] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. “V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation”. In: *Fourth International Conference on 3D Vision, 3DV 2016, Stanford, CA, USA, October 25-28, 2016*. 2016, pp. 565–571.

[62] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan L. Yuille. “The Role of Context for Object Detection and Semantic Segmentation in the Wild”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*. 2014, pp. 891–898.

[63] Peter Ochs, Jitendra Malik, and Thomas Brox. “Segmentation of Moving Objects by Long Term Video Analysis”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 36.6 (2014), pp. 1187–1200.

[64] Anestis Papazoglou and Vittorio Ferrari. “Fast Object Segmentation in Unconstrained Video”. In: *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*. 2013, pp. 1777–1784.

[65] Houwen Peng, Bing Li, Weihua Xiong, Weiming Hu, and Rongrong Ji. “RGBD Salient Object Detection: A Benchmark and Algorithms”. In: *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part III*. 2014, pp. 92–109.

[66] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. “Learning Video Object Segmentation from Static Images”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. 2017, pp. 3491–3500.

[67] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. “Learning Video Object Segmentation from Static Images”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. 2017, pp. 3491–3500.

[68] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc J. Van Gool, Markus H. Gross, and Alexander Sorkine-Hornung. “A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. 2016, pp. 724–732.

[69] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. “The 2017 DAVIS Challenge on Video Object Segmentation”. In: *arXiv* abs/1704.00675 (2017).

[70] Esa Rahtu, Juho Kannala, Mikko Salo, and Janne Heikkilä. “Segmenting Salient Objects from Images and Videos”. In: *Computer Vision - ECCV 2010 - 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part V*. 2010, pp. 366–379.

[71] Hiba Ramadan and Hamid Tairi. “Pattern mining-based video saliency detection: Application to moving object segmentation”. In: *Computers & Electrical Engineering* 70 (2018), pp. 567–579.

[72] Azriel Rosenfeld and John L. Pfaltz. “Distance functions on digital pictures”. In: *Pattern Recognition* 1.1 (1968), pp. 33–61.

[73] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *arXiv* abs/1409.1556 (2014).

[74] Hongmei Song, Wenguan Wang, Sanyuan Zhao, Jianbing Shen, and Kin-Man Lam. “Pyramid Dilated Deeper ConvLSTM for Video Salient Object Detection”. In: *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XI*. 2018, pp. 744–760.

[75] Jia Sun, Dongdong Yu, Yinghong Li, and Changhu Wang. “Mask Propagation Network for Video Object Segmentation”. In: *The 2018 DAVIS Challenge on Video Object Segmentation - CVPR Workshops* (2018).

[76] Yi Tang, Wenbin Zou, Zhi Jin, Yuhuan Chen, Yang Hua, and Xia Li. “Weakly Supervised Salient Object Detection with Spatiotemporal Cascade Neural Networks”. In: *IEEE Trans. Circuits Syst. Video Techn.* (2018).

[77] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. “Learning Motion Patterns in Videos”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. 2017, pp. 531–539.

[78] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. “Learning Video Object Segmentation with Visual Memory”. In: (2017), pp. 4491–4500.

[79] Subarna Tripathi, Youngbae Hwang, Serge J. Belongie, and Truong Q. Nguyen. “Improving streaming video segmentation with early and mid-level visual processing”. In: *IEEE Winter Conference on Applications of Computer Vision, Steamboat Springs, CO, USA, March 24-26, 2014*. 2014, pp. 477–484.

[80] Wei-Chih Tu, Shengfeng He, Qingxiong Yang, and Shao-Yi Chien. “Real-Time Salient Object Detection with a Minimum Spanning Tree”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. 2016, pp. 2334–2342.

[81] Zhengzheng Tu, Andrew Abel, Lei Zhang, Bin Luo, and Amir Hussain. “A New Spatio-Temporal Saliency-Based Video Object Segmentation”. In: *Cognitive Computation* 8.4 (2016), pp. 629–647.

[82] Zhigang Tu, Zuwei Guo, Wei Xie, Mengjia Yan, Remco C. Veltkamp, Baoxin Li, and Junsong Yuan. “Fusing disparate object signatures for salient object detection in video”. In: *Pattern Recognition* 72 (2017), pp. 285–299.

[83] Paul Voigtlaender and Bastian Leibe. “Online Adaptation of Convolutional Neural Networks for Video Object Segmentation”. In: *British Machine Vision Conference 2017, BMVC 2017, London, UK, September 4-7, 2017*. 2017.

[84] Lijun Wang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. “Deep networks for saliency detection via local estimation and global search”. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. 2015, pp. 3183–3192.

[85] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. “Learning to Detect Salient Objects with Image-Level Supervision”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. 2017, pp. 3796–3805.

[86] Linzhao Wang, Lijun Wang, Huchuan Lu, Pingping Zhang, and Xiang Ruan. “Saliency Detection with Recurrent Fully Convolutional Networks”. In: *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*. 2016, pp. 825–841.

[87] Qiong Wang, Lu Zhang, and Kidiyo Kpalma. “Fast filtering-based temporal saliency detection using Minimum Barrier Distance”. In: *2017 IEEE International Conference on Multimedia & Expo Workshops, ICME Workshops, Hong Kong, China, July 10-14, 2017*. 2017, pp. 232–237.

[88] Wenguan Wang, Jianbing Shen, and Fatih Porikli. “Saliency-aware geodesic video object segmentation”. In: *IEEE Conference on Computer Vision and Pat-*

tern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015. 2015, pp. 3395–3402.

[89] Wenguan Wang, Jianbing Shen, and Ling Shao. “Consistent Video Saliency Using Local Gradient Flow Optimization and Global Refinement”. In: *IEEE Trans. Image Processing* 24.11 (2015), pp. 4185–4196.

[90] Wenguan Wang, Jianbing Shen, and Ling Shao. “Video Salient Object Detection via Fully Convolutional Networks”. In: *IEEE Trans. Image Processing* 27.1 (2018), pp. 38–49.

[91] Wenguan Wang, Jianbing Shen, Ruigang Yang, and Fatih Porikli. “Saliency-Aware Video Object Segmentation”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 40.1 (2018), pp. 20–33.

[92] Xiao Wei, Li Song, Rong Xie, and Wenjun Zhang. “Two-stream recurrent convolutional neural networks for video saliency estimation”. In: *2017 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, BMSB 2017, Cagliari, Italy, June 7-9, 2017*. 2017, pp. 1–5.

[93] Tao Xi, Wei Zhao, Han Wang, and Weisi Lin. “Salient Object Detection With Spatiotemporal Background Priors for Video”. In: *IEEE Trans. Image Processing* 26.7 (2017), pp. 3425–3436.

[94] Huaxin Xiao, Jiashi Feng, Guosheng Lin, Yu Liu, and Maojun Zhang. “MoNet: Deep Motion Exploitation for Video Object Segmentation”. In: *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. 2018, pp. 1140–1148.

[95] Change Loy Chen Xiaoxiao Li. “Video Object Segmentation with Joint Re-identification and Attention-Aware Mask Propagation”. In: *The 2018 DAVIS Challenge on Video Object Segmentation - CVPR Workshops* (2018).

[96] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas S. Huang. “YouTube-VOS: A Large-Scale Video Object Segmentation Benchmark”. In: *arXiv* abs/1809.03327 (2018).

[97] Shuangjie Xu, Linchao Bao, and Pan Zhou. “Class-Agnostic Video Object Segmentation without Semantic Re-Identification”. In: *The 2018 DAVIS Challenge on Video Object Segmentation - CVPR Workshops* (2018).

[98] Fanlei Yan. “Autonomous vehicle routing problem solution based on artificial potential field with parallel ant colony optimization (ACO) algorithm”. In: *Pattern Recognition Letters* 116 (2018), pp. 195–199.

[99] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. “Hierarchical Saliency Detection”. In: *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*. 2013, pp. 1155–1162.

[100] Bing Yang, Xiaoyun Zhang, Li Chen, and Zhiyong Gao. “Spatiotemporal salient object detection based on distance transform and energy optimization”. In: *Neurocomputing* 266 (2017), pp. 165–175.

[101] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. “Saliency Detection via Graph-Based Manifold Ranking”. In: *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*. 2013, pp. 3166–3173.

[102] Yun Zhai and Mubarak Shah. “Visual attention detection in video sequences using spatiotemporal cues”. In: *Proceedings of the 14th ACM International Conference on Multimedia, Santa Barbara, CA, USA, October 23-27, 2006*. 2006, pp. 815–824.

[103] Jianming Zhang, Stan Sclaroff, Zhe L. Lin, Xiaohui Shen, Brian L. Price, and Radomír Mech. “Minimum Barrier Salient Object Detection at 80 FPS”. In: *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. 2015, pp. 1404–1412.

[104] Rui Zhao, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. “Saliency detection by multi-context deep learning”. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. 2015, pp. 1265–1274.

[105] Xiaofei Zhou, Zhi Liu, Chen Gong, and Wei Liu. “Improving Video Saliency Detection via Localized Estimation and Spatiotemporal Refinement”. In: *IEEE Trans. Multimedia* 20.11 (2018), pp. 2993–3007.

[106] Xiaofei Zhou, Zhi Liu, Kai Li, and Guangling Sun. “Video saliency detection via bagging-based prediction and spatiotemporal propagation”. In: *J. Visual Communication and Image Representation* 51 (2018), pp. 131–143.

LIST OF FIGURES

LIST OF TABLES

INSA de RENNES
Service des Formations

AVIS DU JURY SUR LA REPRODUCTION DE LA THESE SOUTENUE

Titre de la thèse:
Salient object detection and segmentation in videos

Nom Prénom de l'auteur : WANG QIONG

Membres du jury :
- Monsieur ZHAI Guangtao
- Monsieur LE MEUR Olivier
- Monsieur KPALMA Kidiyo
- Madame ZHANG Lu
- Monsieur COQUIN Didier
- Monsieur DUFAUX Frédéric

Président du jury : COQUIN Didier

Date de la soutenance : 09 Mai 2019

Reproduction de la these soutenue

☒ Thèse pouvant être reproduite en l'état
☐ Thèse pouvant être reproduite après corrections suggérées

Fait à Rennes, le 09 Mai 2019

Signature du président de jury

Le Directeur,

M'hamed DRISSI

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

INSA RENNES

Titre : Détection d'objets saillants et segmentation dans des vidéos

Mots clés : vidéo, détection d'objet saillant, segmentation d'instance d'objet, apprentissage en profondeur

Résumé : Cette thèse est centrée sur le problème de la détection d'objets saillants et de leur segmentation dans une vidéo en vue de détecter les objets les plus attractifs ou d'affecter des identités cohérentes d'objets à chaque pixel d'une séquence vidéo. Concernant la détection d'objets saillants dans vidéo, outre une revue des techniques existantes, une nouvelle approche et l'extension d'un modèle sont proposées; de plus une approche est proposée pour la segmentation d'instances d'objets vidéo.

Pour la détection d'objets saillants dans une vidéo, nous proposons : (1) une approche traditionnelle pour détecter l'objet saillant dans sa totalité à l'aide de la notion de "bordures virtuelles". Un filtre guidé est appliqué sur la sortie temporelle pour intégrer les informations de bord spatial en vue d'une meilleure détection des bords de l'objet saillants.
Une carte globale de saillance spatio-temporelle est obtenue en combinant la carte de saillance spatiale et la carte de saillance temporelle en fonction de l'entropie. (2) Une revue des développements récents des méthodes basées sur l'apprentissage profond est réalisée. Elle inclut les classifications des méthodes de l'état de l'art et de leurs architectures, ainsi qu'une étude expérimentale comparative de leurs performances. (3) Une extension d'un modèle de l'approche traditionnelle proposée en intégrant un procédé de détection d'objet saillant d'image basé sur l'apprentissage profond a permis d'améliorer encore les performances.

Pour la segmentation des instances d'objets dans une vidéo, nous proposons une approche d'apprentissage profond dans laquelle le calcul de la confiance de déformation détermine d'abord la confiance de la carte masquée, puis une sélection sémantique est optimisée pour améliorer la carte déformée, où l'objet est ré-identifié à l'aide de l'étiquettes sémantique de l'objet cible.

Les approches proposées ont été évaluées sur des jeux de données complexes et de grande taille disponibles publiquement et les résultats expérimentaux montrent que les approches proposées sont plus performantes que les méthodes de l'état de l'art.

Title : Salient object detection and segmentation in videos

Keywords : video, salient object detection, object instance segmentation, deep-learning

Abstract : This thesis focuses on the problem of video salient object detection and video object instance segmentation which aim to detect the most attracting objects or assign consistent object IDs to each pixel in a video sequence. One approach, one overview and one extended model are proposed for video salient object detection, and one approach is proposed for video object instance segmentation.

For video salient object detection, we propose: (1) one traditional approach to detect the whole salient object via the adjunction of virtual borders. A guided filter is applied on the temporal output to integrate the spatial edge information for a better detection of the salient object edges. A global spatio-temporal saliency map is obtained by combining the spatial saliency map and the temporal saliency map together according to the entropy. (2) An overview of recent developments for deep-learning based methods is provided. It includes the classifications of the state-of-the-art methods and their frameworks, and the experimental comparison of the performances of the state-of-the-art methods. (3) One extended model further improves the performance of the proposed traditional approach by integrating a deep-learning based image salient object detection method.

For video object instance segmentation, we propose a deep-learning approach in which the warping confidence computation firstly judges the confidence of the mask warped map, then a semantic selection is introduced to optimize the warped map, where the object is re-identified using the semantics labels of the target object.

The proposed approaches have been assessed on the published large-scale and challenging datasets. The experimental results show that the proposed approaches outperform the state-of-the-art methods.